HPVIDEO White Paper

AI Decentralized video-generation

1. Overview

In the era when video has become the mainstream information carrier, AI video generation is accelerating its evolution from "creative toys" to a new generation of content production infrastructure: brand promotion, game previews, educational courses, Meme culture dissemination, KOL personal IP operation, all of which will increasingly rely on AI to automatically produce video content.

However, at present, almost all AI video generation services are built on highly centralized cloud GPU infrastructure (such as AWS, GCP, self-built computer room):-high cost, a large number of "platform premium" superimposed on the price; -Scalability is limited, and the supply of computing power depends on the capital expenditure of a few giants and the pace of computer room expansion; -The access mode is closed, which is extremely unfriendly to automatic call and on-chain settlement, and it is more difficult to support the large-scale use of AI Agent natively.

HPVIDEO was born in response to this structural contradiction.

HPVIDEO is a decentralized AI Video Generation Infrastructure (Video-AI Infra) and AI Agent invocation network deployed on BSC (Binance Smart Chain). By connecting idle GPUs around the world, it builds a high-performance, low-cost, verifiable, trust-free AI video distributed reasoning system.

In the HPVIDEO network:-Any node with an available GPU (PC, Internet cafe, mine, edge server, etc.) can access the network to participate in rendering and receive rewards; -Any developer, project party and AI Agent can submit the video generation task through the on-chain interface; -All task scheduling, payment settlement, node incentive and result verification are automatically executed by smart contracts deployed on BSC, and the rules are open and transparent, without relying on a single platform or company.

Through this design, HPVIDEO transforms the original highly centralized video generation capability into a public computing layer on the open network, allowing "video generation" to evolve from an expensive hosting service to a composable, settable and verifiable infrastructure capability on the chain.

1.1 Three Pain Points of Traditional AI Video Generation

HPVIDEO is designed to address three core pain points in traditional AI video generation: cost, scalability, and Agent adaptation.

(1) Extremely High Cost

Traditional centralized AI video generation typically relies on:-expensive cloud GPUs (such as A100/H100 instances); -High Inference API unit price and bandwidth, storage costs; -The platform is uniformly priced and has a significant "Vendor Premium".

For most Web3 projects, individual creators, and AI agents that need to generate content in batches, this cost structure severely limits the large-scale use of AI video capabilities.

(2) Limited Scalability

Under the centralized architecture, the video reasoning capability is usually limited by:-the GPU supply capability of a single cloud service provider; -Computer room expansion cycle and hardware delivery cycle; -The platform's own business priority (priority to protect the needs of large customers).

Once a large number of Agents or project parties rush to generate demand at the same time, the platform is prone to queuing, flow restriction, allocation reduction and other situations, forming an obvious "single-point bottleneck".

(3) Not Suitable for AI Agent (Not Agent-Native)

Most of the existing video generation platforms are designed with "human creators" as the center:-strong account system, strong KYC, strong manual risk control, relying heavily on the manual audit process; -API permissions are limited, fees are not programmable, and it is difficult to support high-frequency micro-payment and automation strategies; -On-chain settlement and contract invocation experience is poor, and it is difficult to connect with DeFi/Web3 protocol natively.

For fully automated AI agents:-may be banned or restricted at any time due to "abnormal behavior"; -Difficulty in achieving stable large-scale production capacity; -Invocation and payment logic cannot be composable on a chain.

In essence, these systems are "products designed for human creators" rather than "protocols designed for agents".

In the next few years, the main producers of video content will gradually shift from human creators to massive AI agents. This human-centered platform paradigm is naturally not suitable for the future form.

1.2 Overall System Framework of HPVIDEO

In order to solve the above problems, HPVIDEO reconstructs "AI video generation" from the bottom, regards it as a computational task that can be arranged, settled and verified on the chain, and designs the overall system architecture around the following core levels:

1. **Distributed GPU Network**
    - Aggregating global idle GPUs into a unified pool of video inference algorithms;
    - Through node registration, computing power declaration, reputation system and incentive model, GPUs with different configurations are integrated according to

their capabilities.

- Through Frame-Split Rendering and Model Sharding, multi-node parallel rendering and load balancing are realized.

2. On-chain Task Scheduling & Billing

- Use BSC smart contract to manage the task life cycle: submission, bidding, allocation, result confirmation, and reward distribution;

- Use on-chain assets (HPV tokens/BNB) to lock and settle mission costs;

- All rules are written into the contract, open, transparent and auditable, and there is no need to trust a single centralized platform.

3. Composable Video-AI Model Layer

- Support the access of multiple video generation models (Text-to-Video, Image-to-Video, Video Extension, etc.) in the form of "model plug-in";

- Third-party developers are allowed to contribute self-developed models, LoRA and special effects modules;

- Through the unified Model Router, the most appropriate model combination is selected according to the task requirements.

4. AI Agent & Orchestration Layer for Automated Content

- Provide high-level abstract interfaces, such as "Give me a 30-second video of market interpretation";

- The internal layout engine is composed of LLM, script engine, sub-mirror generation, template system, etc.

- Complex video generation is decomposed into a series of low-level rendering tasks, which are completed by GPU nodes in parallel.

5. Multi-node Parallel Rendering and Verifiable Inference

- Frame-level/segment-level slicing is used to split the complete video into subtasks that can be rendered in parallel by different nodes.

- Through redundant rendering (multiple nodes generate the same fragment), hash comparison of results, and ZK-SNARK/ZK-ml introduced in the future, the reasoning results can be verified and calculated.

- Under the premise of not revealing the privacy of model parameters and input, it provides a credible proof for the video generation results.

Through the above levels of collaboration, HPVIDEO reconstructs "AI video generation" from a centralized closed service to a distributed, modular, and programmable on-chain computing infrastructure, which is more suitable for:-Web 3 projects that require continuous output of video content; -Large-scale content farms/UGC platforms; -AI Agent group running on wallet, protocol and transaction system; -DePIN participants who want to gain on-chain benefits through GPUs.

1.3 HPVIDEO's Role in Web3/AI Ecosystem

From the perspective of system role, HPVIDEO is not a single video generation DApp, but the underlying infrastructure serving multiple parties:

- **For Web3 applications and protocols:**
  HPVIDEO is the underlying infrastructure that all applications, protocols and project parties on the chain can call when they need video production, similar to the "decentralized video version OpenAI API" ".
- **For AI Agent:**
  HPVIDEO is a video generation backend designed natively for Agent: it supports automatic billing, automatic settlement, no account, no manual approval, and is the "power system" of video production in the Agent era.
- **For GPU nodes/miners:**
  HPVIDEO provides a new source of revenue: instead of digging traditional PoW, it participates in real AI video reasoning tasks and gets incentives linked to actual computing power contribution.

Thus, from the beginning, HPVIDEO positioned itself as:

> **"Standard Video Generation Layer for Web3 & Agents"**

Rather than a specific ToC application front-end.

2. HPVIDEO's Vision

Over the past decade, video has evolved from a "rich media format" to the default language of the Internet:-Short video instead of graphic social networking; -Live broadcast instead of traditional graphic tutorials; -Games and virtual worlds are gradually being deeply integrated with video content.

In the next decade, driven by AI, this trend will accelerate further-videos will no longer be generated mainly by human creators, but will be continuously and automatically produced by a large number of AI agents.

In this context, the core vision of HPVIDEO can be summarized in three sentences:

1. **Let anyone, any application, any AI Agent, at any time, at a very low cost to generate any video;**

2. **Let the global idle GPU become the common computing base of the video universe.**

3. **Let AI video production change from "service" provided by closed platform to "infrastructure" in open network.**

Around this ultimate vision, HPVIDEO defines long-term goals at three levels:-the infrastructure level; -AI Agent layer; -Economic and governance level.

2.1 Vision of Infrastructure Layer: Become the Global Standard of "video Version of Decentralized Computing Power Network"

At the infrastructure level, HPVIDEO's vision is to:

> Building a sustainable evolving "decentralized AI video computing network" on BSC has become the de facto standard of video AI infrastructure worldwide.

Its core features include:

1. **Decentralization of computing power sources**
    - It does not rely on a few cloud computing giants, but connects multiple computing power sources such as individuals, Internet cafes, mines and edge nodes.

    - Each GPU can join the network through the HPVIDEO Node Runtime;

- Nodes are free to join/leave without compromising the overall availability and security of the network.

2. **On the task and settlement chain**
   - Video generation task is no longer an "internal operation" in the centralized system, but a real transaction on the chain.
   - Task creation, bidding, allocation, completion and reward distribution are all driven by smart contracts.
   - Costs, rules, and priorities are written into the contract, rather than being decided by a company on an ad hoc basis.
3. **Models and capabilities are combinable and scalable**
   - It is not limited to a single video model, but supports plug-in access of multiple models.
   - With the emergence of a new generation of T2V/I2V model, the network can be upgraded smoothly.
   - It has evolved from "API of one model" to "open video computing layer of multi-model collaboration".

In this sense, what HPVIDEO wants to build is not a "cheaper Runway", but a:

> The decentralized version of Runway + AWS + DePIN complex has become the de facto standard of AI video infrastructure under Web3 narrative.

2.2 AI Agent Layer Vision: Become the Video Creation Engine in the Agent Era

If today's video creators are still mainly "human + traditional content teams", then HPVIDEO aims at the next generation of creators: AI Agent Corps.

At the Agent layer, HPVIDEO's vision is to:

> Let AI Agent simply call "generate a video" like calling a function.

This is reflected in several directions:

1. **From "artificial operation platform" to "Agent calling protocol"**
   Traditional platform process: login account → upload material → manually adjust parameters → wait for rendering → download video.
   HPVIDEO wants to evolve into:

```
agent.create_video(
    Prompt = "30-second market analysis video",
    lang="en",
    style="crypto-news"
)
```

A complete video generation and settlement process can be triggered on the chain.

2. **Make Video one of the default output forms for the Agent**
   The output of most agents today is: text, pictures, tables, code.
   HPVIDEO wants to make "video" one of the natural output options for the Agent:
   - Trading strategy Agent → automatically generate strategy interpretation video;
   - Project operation Agent → automatic generation of weekly report/AMA review video;
   - Meme Agent → automatically generate meme video matrix with soundtrack, subtitles and special effects.
3. **Agent system serving various forms**
   HPVIDEO is not tied to any specific Agent framework, but instead serves:
   - Decentralized wallet built-in Agent;
   - Transaction Agent running on the 402Pay/Meme platform;
   - Strategy Agent running in investment systems such as XAI Agent;
   - Multi-Agent system and AI OS built by the third party.

The final ideal state is:

> Countless AI Agents are active on the chain, no longer just sending text and pictures, but continuously generating and publishing videos on HPVIDEO, constantly creating new narratives and attention for users, protocols and markets.

2.3 Vision of the Economic and Governance Level: to Build a Self-Consistent, Sustainable and Multi-Aligned DePIN Ecosystem

At the economic and governance level, HPVIDEO expects to form a self-consistent closed loop:

> Users get video computing power at the lowest possible cost, nodes get returns matching their contributions, developers get sustained benefits, and the protocol maintains long-term sustainable development without excessive centralized operation.

Key designs include:

1. **Let the benefits of GPU nodes be highly linked to the actual contribution**
    - Instead of "undifferentiated mining", they are rewarded by participating in real video rendering tasks.
    - Node revenue is directly related to computing power, online rate and task completion quality.
    - Through the design of incentive function, the nodes with "long-term stable contribution of high-quality computing power" can get the maximum return.
2. **Let model developers and template creators get a share.**
    - Provide model market/template market;
    - Whoever contributes more popular video templates, LoRA and special effects models can get a share in each call.
    - The value of "video content production tools" is returned to the developers and creators who really create value through protocol logic.
3. **Let the agreement governance gradually transit from "founding team leading" to "community co-governance"**
    - In the early stage, the core team led the architecture design and technology landing.
    - With the stability of the network and the growth of users, the key parameters (cost, reward allocation, white list model, etc.) Are gradually handed over to the DAO for decision-making.
    - Through token governance, governance committee and proposal mechanism, decentralized governance can be achieved without sacrificing efficiency.
4. **Forming a positive feedback Flywheel**
    To sum up the economic flywheel that HPVIDEO wants to build in one sentence:

    More users/Agents → More video generation needs → More GPU nodes to join
    → Lower cost and stronger capability → Attract more developers to be compatible →
    More application scenarios
    → Bring in more users/agents again.

Once the flywheel is running, HPVIDEO will no longer be just a single project, but will grow into:-the "operating system" of AI Video DePIN; - "Public Video Computing Layer" for Web3 multi-chain AI applications; - "Video terminal output layer" in the AI Agent economy.

2.4 Long-Term Picture: from "video Generation Protocol" to "video Agent Ecosystem"

In the long run, the goals of HPVIDEO are not only:

> "A decentralized video generation protocol that everyone can call".

Instead, it gradually evolved into:

> "An AI Agent Ecology (Video-Centric AI Agent Ecosystem) with Video as its core expression form".

In this ecosystem:-Video is no longer just passively consumed content, but an important way for Agents to interact with the world; -Each video is a potential entry point for interaction and an expression interface for agents; -HPVIDEO provides the underlying computing power and protocol standards, and truly rich and colorful applications are built by the community and developers.

3. Why Decentralized Video Generation?

To understand why HPVIDEO must be "decentralized" rather than a centralized AI video cloud service, we need to answer a lower question first:

> In the era of AI video, what is limiting innovation? Is it the algorithm itself, or the infrastructure and power structure behind it?

We believe that the real bottleneck of current AI video generation is no longer mainly in the "model effect", but in:-highly centralized cost structure; -Highly centralized supply of computing power; -Access methods and usage rules are determined by a few platforms; -It is difficult for AI Agent to obtain native living space in it.

Therefore, we are not asking "whether to make a video generation tool", but asking:

> Is video, the most important medium of expression, a minority platform or an open network in the future?

This is the question that HPVIDEO wants to answer, and it is also the core reason why we insist on "decentralized video generation".

3.1 Cost Perspective: The Economic Structure of Centralized AI Video Is Unsustainable

The cost structure of traditional centralized AI video services can be simplified as follows:

$$C_{\text{centralized}} = C_{\text{GPU}} + C_{\text{Infra}} + C_{\text{Ops}} + C_{\text{VendorPremium}}$$

Of which:- C_{GPU}: Cloud GPU cost (e.g. A100/H100 hourly); - C_{Infra}: Network, storage, load balancing, operation and maintenance costs; - C_{Ops}Team, risk control, business operation and labor cost; - $C_{\text{VendorPremium}}$: The platform is the premium of profit superposition.

In order to ensure business closed-loop, the platform must ensure that:

$$Price_{\text{user}} \geq C_{\text{centralized}} \times (1 + \text{Margin})$$

This leads to two obvious problems:

1. **A single video is very expensive to generate**
   - For ordinary users: a small amount of use is acceptable;

- For Web3 projects, content factories, and AI agents that require "continuous, large-scale video generation": the cost increases linearly or even superlinearly with scale.

2. **The larger the scale, the higher the platform premium.**
    - When the platform becomes the de facto standard, it has enough motivation to continue to raise fees.
    - AI video computing power has gradually evolved into a "public resource" monopolized by the platform.

The goal of HPVIDEO on the economic model is to rewrite the above formula as:

$$C_{\mathrm{HPVIDEO}} = C'_{\mathrm{GPU}} + C'_{\mathrm{Network}}$$

Of which:- C'_{GPU}: Marginal cost from idle GPUs (close to electricity and depreciation); - C'_{Network}: Decentralized network coordination cost (on-chain Gas + basic development).

By eliminating VendorPremium and most of the Infra/Ops premium, the goal is to achieve:

$$C_{\mathrm{HPVIDEO}} \ll C_{\mathrm{centralized}}$$

In extreme cases, when the network scale is large enough and the number of nodes is large enough, the marginal cost of HPVIDEO can even be close to:

$$C_{\mathrm{HPVIDEO}} \approx C_{\mathrm{electricity}} + C_{\mathrm{hardware\_depreciation}}$$

That is, the cost level close to the "physical limit", which is extremely difficult for any single centralized platform to achieve.

3.2 Scalability Perspective: Centralized GPU Cannot Match the Video Needs of the Agent Era

If the demand for AI video in the future is modeled as:

$$D(t) = D_{\mathrm{human}}(t) + D_{\mathrm{agent}}(t)$$

Of which:- $D_{\mathrm{human}}(t)$Video needs of human creators; - $D_{\mathrm{agent}}(t)$: Video requirements automatically generated by AI Agent.

It is reasonable to expect that with the spread of multi-agent systems, $D_{\text{agent}}(t)$ The growth rate will be much higher than $D_{\text{human}}(t)$, perhaps even close to exponential.

The supply capacity of centralized GPU $S_{\text{centralized}}(t)$ Subject to:-the expansion cycle of the machine room; -Chip production and supply cycle; -Capital expenditure rhythm of a single supplier.

In a high probability scenario, there will be:

$$\lim_{t \to \infty}\left(D(t) - S_{\text{centralized}}(t)\right) \gg 0$$

That is to say, in the future of AI Agent's large-scale emergence, it is almost impossible for centralized platforms to fully match their video generation needs, and they can only be "rationed" through high prices and quota restrictions.

The advantages of the decentralized approach are:-Expanding the supply from "a few data centers" to "all GPU nodes in the world"; -No need for huge construction in the early stage, through economic incentives to drive the natural growth of the network with demand; -Higher upper limit:

$$S_{\text{HPVIDEO}}(t) = \sum_{i=1}^{N(t)} P\,ower_i$$

When the number of nodes $N(t)$ When it is large enough, it can theoretically approach the "total global GPU".

> The centralized platform is supplying water from "a few giant dams", and what HPVIDEO wants to do is "connect all available water sources to the same network".

3.3 Agent Perspective: Centralized Platform Is Not Suitable for AI Agent from the Root.

Typical features of the centralized AI video platform include:-Strong reliance on the account system (account, password, KYC); -Strong manual risk control (behavior monitoring, abnormal ban); -Terms of use often restrict robotic or large-scale automated calls; -The cost model is rigid, and it is difficult to make fine-grained arrangement according to the logic of Agent (such as frequent micro-payment, on-chain settlement, etc.).

For a truly "unattended" AI Agent, this means:-It may be banned at any time for "abnormal behavior"; -It is difficult to obtain large-scale production capacity; -Difficult to integrate seamlessly with on-chain funding/contract logic.

Starting from the Agent requirements, the ideal infrastructure should have:

1. **No need for traditional account system**
    - As long as you have an address on the chain and enough money, you can create a task.
2. **Payment and call logic is highly programmable**
   The Agent can state in the contract or script:
    - When the market fluctuation > X, a 30-second video is generated every 10 minutes;
    - When the popularity of a Meme in social media exceeds the threshold, 10 Meme videos of different styles are automatically generated.
3. **Permissions, fees, and access rules are written on the chain, not in the platform ToS**
    - Everything is bound by public agreements, not by the "subjective judgment" of the platform.

This is exactly what decentralized networks can do and centralized platforms can hardly do:

> HPVIDEO is a protocol designed for agents, not for agents to adapt to a "website designed for humans".

3.4 Power Structure Perspective: Video Production Should Not Be Controlled by a Few Platforms

Video is one of the strongest carriers of modern narrative.

> Whoever has mastered the ability of large-scale video generation, distribution, shielding and censorship will have a huge voice.

Centralized video AI platforms currently tend to have:-Decide what content can be generated and what cannot; -Decide which regions, users and IPs can be accessed; -Decide who to give the API, how much the quota is, and how much the price is; -Ability to modify terms, change rates, block specific users or applications at any time.

In many "ordinary" scenarios, this may seem innocuous; But when it comes to:-sensitive narratives (financial, political, ideological, etc.); -Web 3 projects with transnational circulation; -Relying on fully automated AI Agent economies;

This centralized power will evolve into systemic risk.

HPVIDEO promotes decentralized video generation, not to "oppose regulation" or "pursue anarchy", but to:-transfer the underlying computing power and protocol control from a single company to multiple participants; -Let the rules exist in public smart contracts, not hidden in internal documents; -Avoid the future evolution of AI video ecosystem into a three-in-one super monopoly of "search engine + advertising platform + cloud computing platform".

3.5 Verifiability Perspective: Results Need to Be Trusted, Not Just "trust the Platform"

In a centralized platform, users can usually only "believe" that:-the video is generated by the model claimed by the platform; -No one has tampered with or injected malicious content in the middle; -The inference process is not maliciously simplified (e.g. compression quality, skipping certain steps).

In some key scenarios (visualization of contract execution results, on-chain governance video, compliance education video, etc.), the "authenticity" of the video can not rely solely on a brand Logo.

Decentralized video generation provides room to introduce the following mechanisms:-multi-node redundant rendering (similar to consensus); -Certificate stored on the result hash chain; -Proof of future-accessible ZK-ML/ZK-SNARK inference.

For example, a result can be shown incidentally:

$$\pi = \mathrm{ZKProof}(\mathrm{model,input,output})$$

So that users not only know that "this is generated by a platform", but also know that:-this is inferred from a public version of the model in a certain way; -The reasoning process satisfies verifiable rules, not "black box operations.".

This is crucial for future advanced scenarios such as "machine watching machine video" and "automatic execution of contracts based on video content".

3.6 Resource Utilization Perspective: the Huge Waste of Idle GPUs in the World

At present, there are a large number of idle GPU resources in the world:-graphics card mines after the mining tide recedes; -Internet cafes, computer rooms and office computers that are idle at night; -Game graphics card when the player is not online; -Idle GPUs in various types of edge servers.

These resources are in the state of "eating ash" most of the time, and AI video reasoning just needs a highly concurrent, large-scale and flexible computing pool.

The decentralized video generation network is essentially a:

> A "computing power reuse system" that collects and converts global fragmented GPUs into continuous video computing power.

It can not only greatly reduce costs, but also improve the efficiency of global computing resources allocation, which is a way to optimize social resources from the macro level.

3.7 Summary: The Question Is Not "whether Video Can Be Made", but "to Whom the Video Power Belongs".

To sum up, "Why do we need decentralized video generation" can be condensed into one sentence:

> In the era of AI + Agent, video will become the most important medium of expression and collaboration, and we can not accept that this ability is permanently monopolized by the centralized platform.

HPVIDEO's solution is not to "provide a cheaper video tool", but to answer:-Who controls the production capacity of AI video? -Who owns the infrastructure for video generation? -Who gives the AI Agent the "right" to generate the video?

Our answer is:

> It is decided by the open protocol, the distributed GPU network, and the users, nodes, developers and agents involved in it.

4. Architecture Overview

HPVIDEO reconstructs "AI video generation" into a distributed computing process that can be programmed, settled and verified. It adopts a top-down hierarchical architecture, which is mainly divided into five main layers:

Name of the master layer	Core responsibilities	Key Components/Technologies
Access and Agent Layer	External interaction entrance, providing native support for Agent	Agent SDK、Web DApp、REST/gRPC API
Orchestration and logical layers	Translate high-level requirements into task graphs/Jobs	Prompt Engine、Shot Planner、Model Router、Task Decomposer
On-chain task and settlement layer (BSC)	Trusted Dispatch and Economic Settlement Hub	TaskManager, JobMarket, NodeRegistry, Payment & Reward Contracts
Distributed GPU execution layer	Actual video inference execution	HPVIDEO Node Runtime、Model Executor、Local Verifier、Uploader
Storage and verifiable layers	Video Storage and Result Credibility Assurance	IPFS/Arweave/BNB Greenfield and other off-chain storage, on-chain result hash and ZK

The distributed GPU network, task scheduling on the BSC chain, the AI video model layer, the AI Agent layer, and the orchestration engine are described in more detail below around these main layers.

5. Distributed GPU Network Design

HPVIDEO is committed to building a decentralized video reasoning network composed of idle GPUs around the world, with the core objectives of:

> On the premise of no trust, it efficiently utilizes the fragmentation computing power to provide a stable computing power base for large-scale AI video generation.

5.1 Node Access and Computing Power Modeling

- **Node role:** The GPU execution node is responsible for the actual AI inference task and receives the HPV token/BNB reward.
- **Access process:** The node operator installs the HPVIDEO Node Runtime and completes the registration and pledge on the BSC chain (MIN _ STAKE). The mortgage is mainly used to restrain the evil behavior of nodes.
- **Computing power modeling:**
    - Use uniform indicators $Power_i$ Characterize the first i Comprehensive computing power of nodes;
    - $Power_i$ It is weighted by CUDA core number, available video memory, theoretical floating-point computing power (TFLOPS) and other indicators.
    - $Power_i$ Will be used for task assignment priorities and award assignment weights.

5.2 Task Matching, Reputation and Punishment

- **Task matching:**
    - The matching logic is completed by the contract on the chain and the local Runtime of the node;
    - The node Runtime continuously monitors the Job event on the chain and decides whether to accept the order according to its own resources (video memory, computing power, etc.).
- **Bidding options:**
    - The comprehensive scoring mechanism of "price + reputation weighting" is adopted:

$$Score_i = bid\_price_i - \lambda \cdot Rep_i$$

    - Contract selection $Score_i$ The lowest node, as the winning bidder, rewards the high reputation node while ensuring the economic efficiency.
- **Reputation System (Rep):**

- Dynamically updated, mainly by the mission success rate (S_i), online rate (U_i), average delay (L_i) and other indicators weighted composition;
- The higher the reputation, the higher the order priority and reward rate.

- **Slashing:**
  - For malicious or serious errors such as submitting wrong results and refusing to submit, the agreement will be reduced from the node deposit:

$$Slash_i = \sigma \cdot Stake_i$$

  - At the same time, reduce its reputation value:

$$Rep_i \leftarrow Rep_i \cdot (1 - \delta)$$

Through the above design, HPVIDEO organizes the global fragmented GPU into a unified "AI video supercomputing pool", and realizes task scheduling, billing and anti-evil without trust.

6. On-Chain Task Scheduling on BSC

In HPVIDEO, the BSC plays three roles:-the sole source of task existence; The execution environment for node matching; -Automated ledger for settlement of payments and awards.

6.1 Core Engagement and Mission Life Cycle

The on-chain logic of HPVIDEO is split into the following core contracts:-TaskManager: manages the status, total cost, and subtask list of high-level VideoTasks; -JobMarket: manages the creation, bidding, node allocation, and result recording of specific subtasks (jobs); -NodeRegistry: manage node registration, mortgage, and reputation values; -PaymentAndReward: Manages task cost locking, node reward distribution, and penalty deductions.

A typical mission lifecycle include:

1. **The user submits the task**
    - User/Agent payment $maxFee$ Calling a Task Manager. SubmitTask to create a video task;
2. **Mission disassembly**
    - The scheduling layer under the chain decomposes the high-level task into a plurality of Jobs, calls the JobMarket. CreateJobsForTask to create subtasks on the chain, and records specHash and reward;
3. **Bidding and allocation**
    - The node submits a bid to the Job;
    - The contract selects the node with the lowest score as the winning bidder according to "price + reputation weighting";
4. **Result submission**
    - After completing the reasoning, the winning node uploads the result to the storage under the chain (such as IPFS/Arweave/BNB Greenfield), and writes the resultURI and resultHash back to the chain;
5. **Verification and reward**
    - After passing the verification (Job. Status = Verified), the node can receive the corresponding reward from the PaymentAndReward contract.

6.2 Economic Security and Penalties

- **Reward Formula:**
    - Single Job award R_{job} May be based on node reputation Rep_i Gain bonus:

$$R_{job} = Reward_{base} \cdot (1 + \mu \cdot Rep_i)$$

- **Slashing:**

- Collateral reduction and reputation degradation are triggered when a node submits an erroneous or malicious result:

$$Slash = \sigma \cdot Stake_i, \quad Rep_i \leftarrow Rep_i \cdot (1 - \delta)$$

- **Refund mechanism:**
  - If the task fails or is cancelled, the unconsumed fee can be partially refunded to the user:

$$Refund = totalFee - \sum_{\text{已完成 Job}} R_{job}$$

This on-chain scheduling and settlement architecture ensures the traceability of the whole process of tasks from creation to completion and the economic closed-loop without trust.

7.AI Video Model Layer

The model layer is the core capability layer in HPVIDEO to convert abstract semantic requests into concrete video pixel sequences, and is designed as a set of pluggable, composable, and evolvable model operators.

7.1 Model Space and Request Abstraction

- **Model space (()):**
  Contains a variety of model types, such as Text-to-Video (({ })), Image-to-Video (({ })), style transfer (({ })), subtitle generation (({ })), and so on.

- **Request Space (()):**
  A video request r Is abstracted as a parameter vector:

  $$r = (p, T, R, F, s, l, \kappa)$$

  Amon:

  - p: semantic description (prompt);
  - T: target duration;
  - R: resolution;
  - F: Frame rate;
  - s: Style;
  - l: language;
  - κ: Other optional parameters (such as seed, control strength, LoRA weight, etc.).

- **Model capacity constraints:**
  Each model M Has its own set of capability constraints. $\Omega(M)$, such as maximum resolution $R_{\max}(M)$Maximum frame rate, $F_{\max}(M)$ Etc.

7.2 Cost, Quality and Economic Incentives

- **Model routing:**

  Model Layer Select a model path $\Pi(r)$(I.e., operator combination), will request r Convert to final video v 。

- **Cost Estimate:**

  Call to a single model, cost $C_M(r)$ Roughly proportional to the number of pixels and the length of time;
  The total cost of the path is the sum of the stages:

$$C(\Pi(r)) = \sum C_{M_{i_l}}(r_l)$$

- **Optimizing quality under budget constraints:**

  Given budget B Satisfy $C(\Pi(r)) \leq B$ The goal of model routing is to maximize the integrated quality of the path within the budget. $Q(\Pi(r))$。

- **Model economic incentives:**

  Define a split for each model developer $\rho_k \in [0,1]$, so that its income is:

$$I_k(r) = \rho_k \cdot C_{M_k}(r)$$

By viewing the model as a capability operator, the model layer provides a clear mathematical basis and interface definition for distributed reasoning, task scheduling, and Agent policy optimization.

8. AI Agent Layer

In HPVIDEO, AI Agent is not considered as a "peripheral user", but a first-class citizen in the video generation network.

This chapter focuses on:-Who is initiating the video mission? -Under what conditions? -How to control frequency, budget and style? -How to continuously optimize the strategy based on feedback?

From a formal point of view, HPVIDEO can be regarded as a "video generation operator", while AI Agent is an adaptive policy agent that constantly decides whether to invoke the operator in the time dimension.

8.1 Functional Relationship Between Agent and HPVIDEO

Let:- X : Agent observable state space (such as market, on-chain data, social popularity, historical video performance, etc.); - A : Action space (here refers to "whether to generate a video and what kind of video to generate"); - V The video space (the abstract object pointed to by the video file or its URI) that the HPVIDEO network can output.

HPVIDEO is abstracted as a video generation operator:

$$G:\mathfrak{R} \to V$$

Wherein the request spaces $\mathfrak{R}$ As described above.

At discrete time step t , Agent observed state $x_t \in X$? Select the action $a_t \in A$ A typical type of action is "initiate a video task", that is, construct a request. r_t:

$$a_t = \text{"create\_video"}(r_t)$$

The Agent's interaction with the HPVIDEO can then be formalized as:

$$\begin{aligned} r_t &= \pi_\theta(x_t) \\ v_t &= G(r_t) \end{aligned}$$

Of which:- π_θ: By parameter θ The controlling Agent policy function maps the state to a video request; - v_t: At the moment t Video results returned by HPVIDEO.

8.2 Triggers and Templates: From Policies to Task Templates

In practical use, AI Agent usually does not "construct a video request from scratch" at every step, but calls it based on a predefined task Template.

A task template can be represented as a triple:

$$T = (\Gamma, S, D)$$

Of which:- Γ : Trigger condition, which determines "when to initiate the video generation corresponding to the template"; - S : Static configuration, such as video length, resolution, style, etc.

- D A collection of data sources used to populate content (such as market data, news summaries, on-chain events, etc.).

Trigger condition Γ It can be further subdivided into:

1. Time-based: for example, "Video is generated once every day at 18:00";
2. Event-based: for example, "a video is generated when the price breaks through a certain threshold";
3. Condition-based trigger: for example, "trigger when TVL increases by more than 30% within 24 hours and social popularity doubles".

When the moment t Meet a certain template T The trigger condition for $\Gamma(t) = 1$ The Agent, based on the template definition, retrieves data from the data source D Extract the latest data and construct the request parameter r_t, and then call HPVIDEO to generate the video.

8.3 Budget & Payment Constraints

AI Agent calls HPVIDEO must be done under budget constraints to avoid unconstrained consumption of funds.

Let:- B_{day}: Agent daily budget ceiling; - c_t: At time t Cost of video tasks initiated; - $C_{day}(d)$: Someday d Cumulative cost and.

Hav:

$$C_{day}(d) = \sum_{t \in T_d} c_t$$

The budget constraint is:

$$C_{day}(d) \leq B_{day}$$

When a cost is expected to be generated c_t Make:

$$C_{day}(d) + c_t > B_{day}$$

The Agent should either reject this build, or use a "degraded parameter" (reduced resolution/duration/frame rate) to reduce the c_t。

In a more general case, you can also define a medium- to long-term budget B_{total}, and requires:

$$\sum_{d=1}^{D} C_{day}(d) \leq B_{total}$$

8.4 Reward & Objective of Agent

The goal of Agent is not to "generate as many videos as possible", but to generate a collection of videos that brings the greatest comprehensive benefits.

Let:- v_t: At time t The generated video; - m_t: Performance indicators of the video in a period of time (such as number of plays, completion rate, number of likes, number of retweets, click-through rate, etc.); - R_t: Proceeds from this video.

A payoff function can be defined:

$$R_t = F(m_t)$$

For exampl:

$$R_t = \alpha \cdot \mathrm{views}_t + \beta \cdot \mathrm{likes}_t + \gamma \cdot \mathrm{shares}_t$$

After considering the costs, you can define the net benefit:

$$\widetilde{R}_t = R_t - \lambda \cdot c_t$$

Among λ Is the cost penalty weight.

The long-term optimization objective of the Agent is to maximize the expected cumulative net revenue over a certain time horizon:

$$\max_{\theta} \mathbb{E}\left[\sum_{t=1}^{T} \widetilde{R}_t\right]$$

8.5 Policy Update & Adaptation

In practice, the Agent can dynamically adjust the policy parameters according to the historical data θ 。

Abstractly, one policy update can be written as:

$$\theta_{t+1} = \theta_t + \eta \cdot \nabla_{\theta} J(\theta_t)$$

Of which:- η : Learning rate; - $J(\theta)$Objective function (e.g., estimation of long-term cumulative net income); - $\nabla_{\theta} J(\theta_t)$: estimated gradient direction based on historical video representation.

From a macro perspective, over time:-the Agent adjusts the frequency of generation (when to send the video); -Adjust the structure and style of video content (what to say and how to say it); -Resource allocation will be adjusted (in which scenarios are you willing to invest more to generate high-quality video).

8.6 Overall Constraints for Agent – HPVIDEO Interaction

To sum up, the behavior of AI Agent layer on HPVIDEO can be regarded as a dynamic optimization problem under the following constraints:

1. **Resource Constraints (Budget):**

$\forall d,\ C_{day}(d) \le B_{day}$

2. **Invocation constraints (trigger conditions):**
   Only when $\Gamma(t) = 1$ The Agent can initiate a video request under a certain template.

3. **Agreement Constraints (Compliance and Permissions):**
   There exists a feasible strategy space Π_{valid}? Requirements $\pi_\theta \in \Pi_{valid}$, for example:

   - Some restricted models are not allowed to be called;
   - The generation of a specific type of content at a specific address or region is not allow.

4. **Target optimization:**
   Subject to all the above constraints, maximize:

$$\mathbb{E}\left[\sum_{t=1}^{T} \widetilde{R}_t\right]$$

8.7 Summary

From a mathematical perspective, the AI Agent layer of HPVIDEO is essentially:

> Under a given budget and policy constraints, a video generation operator is generated by repeatedly calling G The sequential decision system of "whether to generate, what to generate and when to generate" is continuously carried out in the time dimension, and the strategy parameters are adaptively updated through feedback signals.

The value provided by HPVIDEO for this system is:-Video operators G Is low cost, highly scalable, and verifiable; -The calling and charging are completely chained and naturally adapted to the use of the Agent; -The structure allows the Agent to automatically generate video on a large scale in the future without relying on a centralized platform interface.

9. HPVIDEO Orchestration Engine

The orchestration engine is the central brain of HPVIDEO between "upper demand" and "bottom computing power". It is responsible for sending a high-level request:

> "Generate a 30-second 1080p weekly market report video, bilingual in Chinese and English, with a partial news broadcast style."

Disassemble into a series of atomic tasks that can be executed in a distributed GPU network, and ensure that:-visual, audio, subtitles and other multimodal parts are logically consistent; -The cost is within the budget; -Duration, resolution, style, etc. Meet expectations.

9.1 Abstract Perspectives: From Semantics to Task Graphs

At the formal level, the orchestration engine can be viewed as a mapping:

$$\Phi : \mathfrak{R} \rightarrow \mathcal{G}$$

- $\mathfrak{R}$: high-level "Requests", including prompt, duration, resolution, language, style, budget, etc.;
- $\mathcal{G}$: Task Graph space, each graph $G = (V, E)$ Is a directed acyclic graph (DAG).

Where:-set of vertices V : Various operators to be executed in the video generation process (script generation, T2V, frame insertion, style migration, subtitle, audio, etc.); -Edge set E Data flows and dependencies between operators.

Each vertex $v_i \in V$ Corresponding to an operator call and its argument (O_i, θ_i)。 What the orchestration engine does is construct such a graph. G, and guarantee that:-all dependencies are in legal order (acyclic); -the total inferential cost does not exceed the budget; — The output meets user-specified quality and duration constraints.

9.2 Multi-Objective Optimization of Cost and Quality

For a video task, the task map G Corresponding total cost:

$$C(G) = \sum_{v_i \in V} C_i$$

Among C_i It's an operator v_i The estimated cost of, which can usually be written as:

$C_i = k_i \cdot R \cdot F \cdot T_i$

- R: The number of pixels corresponding to the resolution (e.g. $R = width \times height$) ;
- F: Frame rate;
- T_i: length of time the operator acts (seconds);
- k_i: a constant related to the complexity of the concrete model.

The overall quality can be abstracted as:

$$Q(G) = \sum_{v_i \in V} w_i \cdot Q_i$$

Among w_i It is the weight of different stages in subjective quality.

The orchestration engine is faced with a constrained multi-objective optimization problem:

$$\max_G Q(G) \quad \text{s.t.} \quad C(G) \leq B$$

When the budget is low, the orchestration engine enforces a "downgrade strategy", such as:-Reduce the resolution R ; -Reduce the frame rate F ; -Shorten the duration T ; -Or choose a more "lightweight" model (smaller k_i) 。

9.3 Timeline & Shot Segmentation

Video is essentially a continuous signal on a time axis. The orchestration layer first cuts the entire video into several non-overlapping segments on the timeline:

$$[0,T] = \bigcup_{j=1}^{M} [\tau_{j-1}, \tau_j]$$

- Total duration T Usually specified by the user or Agent;
- $\{\tau_j\}$ Is a segment boundary, satisfying $0 = \tau_0 < \tau_1 < \cdots < \tau_M = T$。

Each segment corresponds to a shot, which can be expressed as:

$$S_j = (\Delta T_j, \mathrm{style}_j, \mathrm{prompt}_j, \mathrm{model}_j)$$

Of which:- $\Delta T_j = \tau_j - \tau_{j-1}$ Is the length of the shot; - style_j: visual style tag; - prompt_j: semantic description; - model_j: The master model ID used to generate the fragment.

The goal of shot segmentation is to:-align semantically with the script structure (paragraphs, scene changes); -Technically easy to render in slices (different shots can be generated in parallel by different nodes).

9.4 Task Graph to Job Segmentation (Task Graph → Jobs)

Logical layer task map G It needs to be further divided into Job sets that can be sent down to the GPU network:

$$J = \{J_1, J_2, \ldots, J_N\}$$

Each Job J_n Corresponding to:-a shot interval (or frame interval); An operator (or combination of operators); -Device capabilities to be met (lower limit of video memory, computing power requirements, etc.).

Yes, first j Shot, possibly split into:-Main T2V Job: Generate base video; -Frame Insertion Job: Increase the frame rate; -Style Job: perform style migration; -Subtitle Job/Audio Job, etc.

Ruodi j Corresponding to a lens k_j Jobs, the total number of jobs is:

$$N = \sum_{j=1}^{M} k_j$$

Each Job J_n Comes with specifications (resolution, frame rate, duration), estimated cost C_n And dependent information.

9.5 Parallelism & Granularity

The orchestration layer can control the "task granularity", that is, the time length and complexity of each Job:-the granularity is too large → the execution time of a single Job is too long and the load is uneven; -The granularity is too small → the scheduling overhead and network transmission cost are too high.

By observing the performance of historical tasks, the orchestration engine will dynamically adjust the job granularity to achieve a balance between scheduling overhead and load balancing.

9.6 Pipeline Topology

General video generation uses a multi-stage pipeline, which can be abstracted as a multi-layer directed acyclic graph:

- Stage 1: Basic video generation (low frame rate/basic style);
- Stage 2: Time domain enhancement (frame insertion, slow motion, etc.);
- Stage 3: Style/Effects and Subtitle Overlay.

Record the output of each stage as $V^{(1)}, V^{(2)}, V^{(3)}$, then:

$$\begin{aligned} V^{(1)} &= f_1(P, \Theta_1) \\ V^{(2)} &= f_2\left(V^{(1)}, \Theta_2\right) \\ V^{(3)} &= f_3\left(V^{(2)}, \Theta_3\right) \end{aligned}$$

Final output video:

$$V_{final} = V^{(3)}$$

Among P For raw prompt and script information, Θ_l Is the parameter of each stage.

9.7 Budget-Aware Orchestration

The user or Agent gives an upper budget limit before the video is generated B The orchestration engine needs to ensure that:

$$C(G) = \sum_{n=1}^{N} C_n \leq B$$

When the pipeline is initially planned G_0 When the budget is exceeded, the orchestration engine looks for a mapping in the Downgrade Policy space T_s, such that:

$$G' = T_s(G_0)$$

Satisfied:- $C\left(G'\right) \leq B$; - $Q\left(G'\right)$ As close as possible $Q(G_0)$。

Common degradation methods include:-resolution scaling; -Frame rate scaling; -reduced duration; -replace with a lighter model, etc.

9.8 Interfacing Orchestration to Distributed Execution

The core information that the orchestration engine finally submits to the on-chain and GPU network includes:-Job list J : Specifications, estimated costs, and dependencies for each Job; -Task dependency structure: describe execution order constraints via "predecessor sets":

If $J_a \in Pred(J_b)$, you must ensure that J_a At J_b Execute before.

As long as the execution layer abides by the above dependencies, the overall video generation process can be completed in an orderly manner.

9.9 Summary

HPVIDEO's orchestration engine transforms high-level video requirements into a budget-aware, multi-stage, multi-operator, parallelizable task graph, which is further split into a set of Jobs that can be executed in a distributed GPU network, with formal constraints:-Total cost $C(G)$ Do not exceed the budget; -Quality $Q(G)$ Maximize as much as possible under constraints; Job granularity and parallelism are balanced between scheduling overhead and load balancing; -The topology of the multi-stage pipeline maintains clear forward-backward dependencies.

10. Roadmap

This section outlines the technical and ecological evolution path of HPVIDEO from the time dimension, covering Q1 – Q4 (first year) and Y2 – Y3 (second and third year).

Q1 – MVP Phase: Underlying Network and Foundation Capability (MVP)

Technical Milestones:

1. **Distributed GPU Network Launch (HPVIDEO Node v1)**
   - Release the first version of HPVIDEO node runtime (support Docker/binary form);
   - Support basic hardware self-test (CUDA/video memory/bandwidth) and on-chain registration and mortgage;
   - Complete the small-scale internal test network of 100 + nodes, and get through the complete closed loop of "registration → receiving orders → execution → receiving rewards".
2. **SVD-XT Model Integration (AI Video Backbone Capability)**
   - Integration of Stable Video Diffusion/SVD-XT as the first backbone Text-to-Video model;
   - Support 720p, short-duration (5 – 10 seconds) video generation, and adapt to HPVIDEO model abstract interface;
   - Establish the basic cost estimation function and complete the performance benchmark test of SVD-XT on different types of graphics cards.
3. **BSC Task Scheduling Contract Deployment (Task & Job Contract v1)**
   - Deploy the first version of Task Manager/JobMarket/Node Registry/Payment contract in BSC;
   - Support the processes of on-chain registration, node order receiving and result submission with a single task split into multiple Jobs;
   - Complete internal test of test network and small-scale main network to verify Gas cost and task throughput capacity.

Product and Ecology:-Release of HPVIDEO Web Console (Alpha): Support ordinary users to manually create video tasks, view execution status and results; -Complete the access of the first batch of early cooperation projects (such as encrypted project promotional videos and daily market videos).

Q2 – Agent Evolution Phase: From "Human Operation" to "Agent Invocation"

Technical Milestones:

1. **Video Agent SDK (Python/Type Script)**
    - Defining the Unified Request Parameter Space $\mathfrak{R}$ Encapsulate the interaction details on the chain;
    - AI Agent is allowed to initiate video tasks through simple function calls and pay automatically within budget constraints.
    - Support basic budget control (daily limit), simple retry strategy and result callback.
2. **Template Marketplace (v1)**
    - Introduce the concept of "task template": fixed video structure + pluggable data source;
    - The first batch of official templates: daily quotation, weekly review, project update, Meme short video, etc.;
    - Support the third-party template creator to upload the template, and reserve the chain identification (template ID) for the subsequent template sharing mechanism.
3. **Multi-node frame rendering (Multi-node Frame Rendering)**
    - Dividing the video time axis into a plurality of segments, and rendering the segments on different nodes in parallel;
    - Get through the logic of segment synthesis, timeline alignment and audio synchronization to ensure visual continuity;
    - Complete stress testing at 1,000 + node scale to validate throughput and latency.

Product and ecology:-Launch the "Agent Video Creator Program" to encourage project parties/developers to build their own agents (market broadcast Agent, Meme Agent, operation Agent, etc.) around HPVIDEO; -Integrate at least 3 external AI Agents/wallet products to call HPVIDEO directly from wallet/Bot to generate video.

Q3 – Mass Adoption Phase: Real-World GPU and Multimodal Capabilities

Technical Milestones:

1. **Cybercaf Cybercafé Program**
    - Design the node deployment scheme for the Internet bar scenario (batch installation, remote operation and maintenance, bandwidth and time period strategy);
    - Batch cooperation with Internet cafes/computer rooms in key areas (such as GPU-intensive areas in Southeast Asia and China);
    - Through the differentiated rate and long-term benefit model, the GPUs during the idle period of Internet cafes (night/early morning) are leased to the HPVIDEO network.
2. **10,000 + GPU Node Scale (Network Scale-Up)**

- The total number of network nodes is expanded to 10,000 +, covering multiple regions and operators (individuals, Internet cafes, mines, edge nodes, etc.);
- The node reputation system and that schedule algorithm are optimized, so that the high-quality node can obtain higher order receiving priority and income;
- A stable global distributed video computing pool is formed to provide guarantee for long-term, high-resolution and batch tasks.

3. **AI Multimodal Video Model Support (Multimodal Video Models)**
   - On the basis of the original T2V capability, a multimodal video generation model supporting "text + image + structured data" is introduced.
   - Support I2V generation based on Logo/role settings/UI screenshots and other visual elements, as well as mixed text + data-driven video;
   - It provides the integrated generation capability of "chart + explanation + dynamic effect" for complex scenarios such as market interpretation, tutorial video and data visualization video.

Q4 – Protocol Deepening Phase: Verifiable Computing and Model/Template Economics

Technical Milestones:

1. **Verifiable Inference and Result Validation Enhancement (Verifiable Inference v1)**
   - A multi-node redundant rendering and hash consistency checking mechanism is introduced for key tasks;
   - Record more fine-grained task execution metadata (participating nodes, computing power consumption estimation, result hash, etc.) on the BSC chain;
   - ZK-ML/ZK-SNARK interface design is reserved to lay the foundation for subsequent zero-knowledge video reasoning proof.
2. **Model & Template Economics v1**
   - The design and implementation of the model registry and the template registry on the chain are completed.
   - Support to split the cost of each task: node reward, model author share, template author share, and agreement treasury;
   - Preliminary online configurable split parameters (e.g. node 70 – 80%, model + template 10 – 20%, protocol vault 5 – 10%).
3. **Node Reputation System v2 (Reputation 2.0)**
   - Upgrade from simple "success rate + online rate" to multi-dimensional reputation model:
     - Response speed of receiving orders;
     - Complicated task success rate;
     - Video quality feedback;
     - Controversy rate, etc.;

- Reputation score is directly linked to task priority and reward multiplier, forming a positive incentive of "the busier the good node, the more it earns".

Products and Governance:-Governance Alpha
-Establish the agreement treasury and basic governance framework, and introduce the basic process of "proposal-voting-execution"; -Migrate some on-chain parameters (such as reward distribution ratio, minimum pledge, black and white list rules) to governance control; -Invite early nodes, model developers and ecological partners to participate in the "governance test season".

Y2 – Global Expansion & Multichain

Assuming that Q1 – Q4 completed the "0 – 1 phase" of the HPVIDEO network, the goal of the second year (Y2) is to expand from single-chain and single-ecosystem to a global AI video infrastructure with multi-chain, multi-scene and multi-partner collaboration.

Technical direction:

1. **Multichain & Cross-chain Settlement**
    - On the premise of keeping BSC as the main settlement layer, explore the lightweight deployment scheme on other EVM compatible chains;
    - Introduce cross-chain bridge or cross-chain message to support initiating video task on other chains and completing settlement on BSC;
    - To achieve "where the user is in the chain, where the Agent is in the chain, HPVIDEO will provide video computing power for it in the chain".
2. **Advanced Multimodal Models and Charts/Data Visualization Video (Advanced Multimodal Video)**
    - Deep support for multi-modal video generation of "text + image + table/time series data";
    - Automatically generate market interpretation video with charts and visual video of contract execution/DAO voting results;
    - Add "structured data channel" to the model abstraction, so that the video is not only "good-looking", but also "information density".
3. **Large-scale scheduling optimization and SLA (Scheduler v2 + SLA)**
    - A scheduling strategy is trained based on historical task data, so that high-priority tasks can still obtain guaranteed delay when the network is congested;
    - Introduce the concept of task SLA (for example, in 95% of cases, 30 seconds of video is completed within X minutes);
    - Through the reward/penalty mechanism, the node behavior is strongly bound to the SLA objective.

Ecological direction:

1. **Vertical Solutions**

- Provide complete industry solutions for several main battlefields:
  - Web 3 project operation: market + community + governance video automation;
  - Education/Tutorial: Product Onboarding/Functional Instructional Video Automation;
  - Meme ecology: daily update meme short video matrix.
- Form a joint case of "industry X HPVIDEO" "to promote large-scale access of B-terminal.

2. **DePIN Alliances**
   - Collaborate with other computing power, storage, bandwidth and AI model networks to form a DePIN alliance;
   - Create a unified narrative: HPVIDEO = "AI video layer", and build a "modular AI infrastructure stack" with other protocols.

Y3 – Standard Video Layer for AI OS & Agents

The goal of the third year (Y3) is to upgrade HPVIDEO from "a powerful AI video DePIN network" to:

> The "video output layer standard" called by various AI OS/super Agent platforms by default.

Technology and Protocol Evolution:

1. **Deep integration with mainstream AI OS/Agent platform**
   - HPVIDEO is abstracted as a standardized video output interface, so that various AI OS and Agent frameworks can be used out of the box.
   - Different Agents can automatically call HPVIDEO in the background only by following the agreed parameter structure.
   - Drive market perceptions like "OpenAI for text", "XXX for image", and HPVIDEO = "for video".
2. **ZK-ML Video Proof of Reasoning v2 (optional)**
   - Provide ZK-level reasoning proof capability for videos with high value and high credibility requirements (contract description, compliance explanation, legal scenarios, etc.);
   - Support the "third-party verifier" to verify the video generation process at the mathematical level, so that the credibility of "machine understanding video" no longer depends solely on the centralized brand.
3. **Modular protocol split and pluggable component**
   - HPVIDEO is disassembled into a plurality of protocol modules which can be independently evolved:
     - Task scheduling module;
     - Model market module;

        - Node network module;
        - Agent calling and template module;
    - Allow other projects to integrate only some of the modules, not the entire system.

Governance and community:

1. **Mature DAO Governance and Parameter Adaptation**
    - Migrate more key parameters to DAO decisions, including:
        - Share proportion;
        - Reputation algorithm weight;
        - Model/template whitelist policy;
        - Incentive plan, resource allocation, etc.
    - The introduction of a data-based "parameter-adaptive proposal system":
        - The DAO automatically generates a plurality of parameter adjustment candidate schemes according to the running data on the chain;
        - The members of the community vote on the candidates.
2. **Developer and Creator Eco Flywheel**
    - Form a clearly layered ecological role:
        - GPU nodes provide computing power;
        - The model developer/template author provides authoring tools;
        - Agent developers construct various video agents;
        - Project party/creator/end user consumes video output;
    - Reward the most active and innovative participants in HPVIDEO through annual awards, grant programs, hackathons, etc.