On this page

- Current status
- Network
- Stage 1 - Beta
- Stage 2 - Multimodal
- Stage 3 - Synthetic data
- Stage 4 - Auto-balancing
- Stage 5 - Sharding
- Stage 6 - LoRA / SFT
- Stage 7 - RL
- Stage 8 - Full-Parameter SFT
- Stage 9 - Training suite
- Stage 10 - Pre-training
- Dolphin models
- SDK for games
- Custom bots
- Dolphin Dev

Roadmap

Current status and achievements

Archived copy · cryp2nova.com

1. We created the default model for all Venice.ai users -Dolphin Mistral 24B Venice Edition.
2. Our models now exceed five million monthly downloads on Hugging Face.
3. Dolphin Inference Network beta is now live (https://x.com/dphnAI/status/1978158402431041657)

1 - Dolphin Network - Distributed AI Development Network

Stage 1 - Distributed Inference Beta

We have launched a distributed AI inference network designed to repurpose idle GPU compute from consumers and idle data centers. Unlike session-based rental marketplaces such as Akash or io.net, our peer-to-pool architecture lets operators join and leave at any time without reservations or penalties, aligning with the needs of users who frequently reclaim their GPUs.

To ensure model integrity, we sample log probabilities and related signals to fingerprint outputs and verify that nodes are running the expected models and configurations. This approach discourages model substitution, excessive compression, or fabricated outputs. The network has already been used to generate synthetic data for future Dolphin models. Additional details are available in the announcement thread on X.

Stage 2 - Image / video / audio / transcription nodes

Over the next one to three months, with the release of version 2 of the node software, the network will expand to host popular open-source models across text, image, video, audio, and transcription (for example, Whisper). Nodes will also receive a local API for priority access to the model running on their machine, encouraging local AI users to share excess capacity with the Dolphin Network.

Stage 3 - Synthetic data generation suite

The network will support a synthetic data generation suite that leverages queued jobs on idle network capacity to reduce per-token costs for non-urgent workloads. Powered by our latest large models, including Dolphin X1 235B, it will deliver high-quality synthetic datasets at a fraction of closed-source costs, with support for uncensored training data, fiction, and other creative needs.

Stage 4 - Lighthouse auto-balancing

Based on inference demand, nodes shift between mostly generating inference requests during periods of high network demand, and processing less time-sensitive tasks such as Data generation, RL & training while idle

A busy day might look like 90% of nodes processing inference, and 10% of nodes handling fine-tuning/RL/data generation

Archived copy · cryp2nova.com

Whereas if inference demand was lower, instead of nodes going underutilized, the lighthouse will swap them to training

Stage 5 - Sharded distributed inference



Right now, our nodes run a single small or medium sized model - with size depending on their GPU. We are developing pipeline parallel model sharding to distribute one large model across dispersed GPUs, for example, four RTX 4090s in different locations, each serving one quarter of the model. To our knowledge, fault-tolerant sharded inference has not been demonstrated in production; moving quickly could make us the first.

Stage 6 - Distributed LoRA / SFT (Supervised Fine Tuning)

Enable distributed fine-tuning on Dolphin Network. Start with LoRA adapters - tiny updates, short runs, broad hardware reach - to prove stability and unit economics on consumer links. Workers train locally for many steps then send compact changelogs instead of full tensors. Lost detail is tracked via error-feedback buffers and periodic anchoring, keeping sync payloads in the KB range.

Stage 7 - Distributed Reinforcement Learning

Our sharded inference engine will be used to power a distributed reinforcement learning platform, with large models powering both rewards / verifiers and generating rollouts for LoRA reinforcement learning runs on our network. Small size of LoRA adapters allows them to be rapidly distributed among network workers. Quality of LoRA for reinforcement learning is comparable to full finetuning according to research by Thinking Machines

Stage 8 - Distributed Full-Parameter SFT

Extend the network to run supervised fine-tuning across participating nodes. Research about adding intra-worker sharding (FSDP/TP/PP) where possible. If we can achieve this, we would be the first protocol to train large models in a split system (each node gets a shard of the model)

Stage 9 - Model creation & fine-tuning suite

A user-friendly interface to submit a training run powered by our network, with support for SFT, RL and synthetic dataset generation. Users can upload their own datasets, select from our curated options, or combine both. The primary objective is to provide training infrastructure at a lower cost than major cloud providers, while also serving the no-code market by allowing users to train models on custom data without prior experience.

Stage 10 - Large scale Distributed Pre-training

Archived copy · cryp2nova.com

Exploratory R&D to evaluate distributed pre-training on pooled idle GPUs. We will prototype small-scale runs using fault-tolerant sharded training, streaming data pipelines, and frequent checkpointing. Expansion beyond prototypes depends on quality/cost results relative to centralized alternatives and on network stability in prior stages.

2 - Dolphin Models

Dolphin is trained on top of leading open-source frontier models with the goal of allowing users to have control over the model and its alignment.

Dolphin 24B Venice Edition, trained on Mistral Small 24B is available in our chat UI at chat.dphn.ai - https://x.com/AskVenice/status/1917001067071410661

Dolphin X1 8B, trained on Llama3.1 8B is available in our Telegram bot @DolphinAI_bot - https://x.com/dphnAI/status/1977769183992852835

Dolphin X1 235B, trained on Qwen3-235B and Dolphin X1 405B, trained on Llama-3.1-Tulu-3-405B - our most powerful upcoming models.

Model lineup

- Dolphin X1- uncensored by default while keeping the base model's skills and intelligence.
- Dolphin RP - creative, role-playing models suited for personas and storytelling.
- Dolphin S1 - steerable alignment level via request-time flags.

Dolphin inference for game developers

The Dolphin Inference SDK will enable game developers to power NPC dialogue and other in-game AI.

These models are optimized for strong system-prompt adherence with reduced bias and model restrictions, making them well suited for RPGs and role-playing experiences that require nuanced character behaviors beyond the limits of many centralized LLM APIs.

While local inference is possible, we recommend our managed API to minimize client-side VRAM usage and simplify integration.

CRYP2NOVA

Archived copy · cryp2nova.com

Custom Telegram and Discord bots

A platform for creating AI-powered Telegram or Discord bots backed by Dolphin Network - similar to character.ai but for bots.

- Choose text and image models, set name, character description, and avatar.
- Bots can respond instantly or at randomized intervals.
- Configure a character seed for consistent, on-style image generations over time.

Read more here dphn.ai/bot and try the Dolphin Telegram bot @DolphinAI_bot.

Dolphin-dev

A data-generation system focused on problem solving with tools inside disposable environments. Most open-source datasets teach pattern matching rather than planning, tool use, verification, and iteration. Dolphin Dev addresses this by deploying LLMs into sandboxed environments to solve challenges with tools, producing datasets that improve our existing model lines and pave the way for an autonomous coding product.

This will make our models performant for coding assistance, in applications such as Cline, Cursor, Roo, etc.

With these datasets, we can train dedicated coding models that run autonomously inside worker-deployed containers to complete jobs. A future web interface will allow submitting a job; after completion, the worker returns the work product and logs.

Worker-hosted sandboxes (opt-in)

Archived copy · cryp2nova.com

Nodes on Dolphin Network can dedicate a slice of RAM/CPU to host disposable sandboxes for these jobs. The sandbox and co-located LLM iterate locally - plan → call tools → run tests → refine - so only the final artifact and logs are sent back to the customer. This reduces latency, increases iterations per minute, and monetizes idle CPU/RAM alongside GPU.

Archived copy · cryp2nova.com

Archived copy · cryp2nova.com