rebellions_

Breaking Barriers in Physical AI:

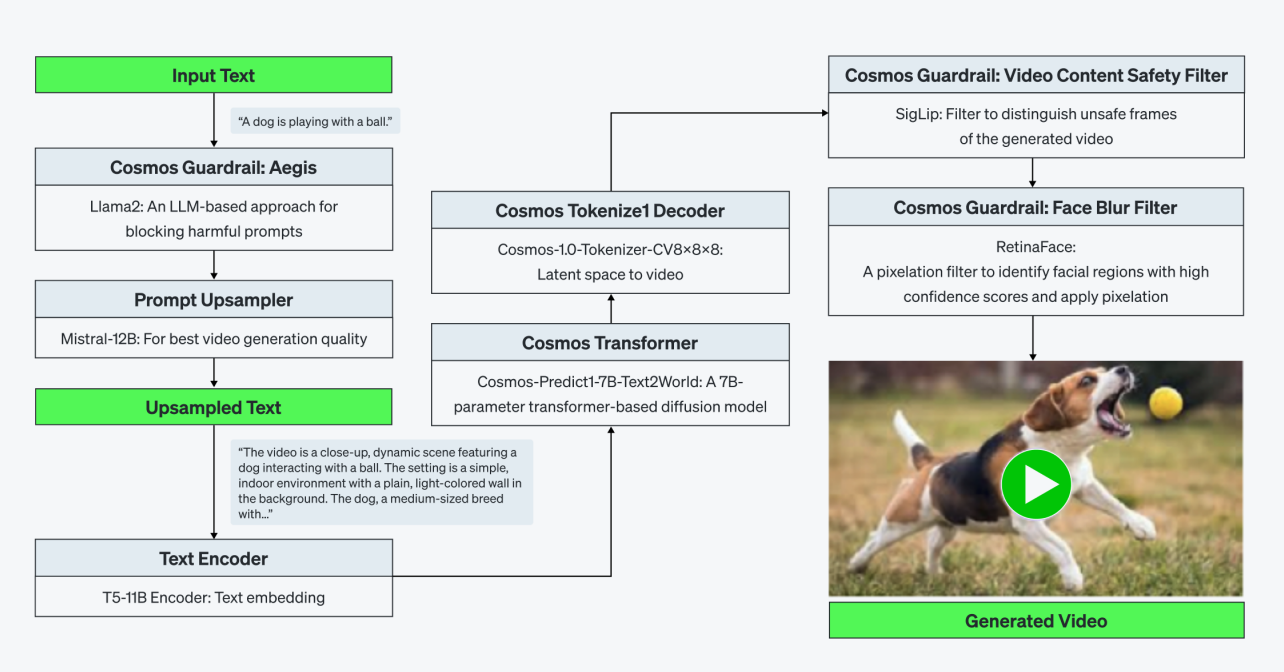
Cosmos Runs on ATOM™

May 19, 2025

Multimodal AI is redefining the limits of infrastructure. Models are no longer confined to single input types—they now span text, images, video, and more. At the frontier of this shift is Cosmos*: a diffusion based text-to-video generation model that integrates at least 5-6 distinct AI subsystems, including tokenizer, pre-trained world foundation models, and guardrail system (see Figure 1). Each subsystem brings its own architectural complexity and compute behavior.

Traditionally, such workloads have been bound to high-end GPU platforms—not because they are ideally suited, but because nothing else could handle the scale. GPUs remain powerful general-purpose accelerators, but their reliance on fixed software stacks and rigid memory hierarchies often make adaptation slow, inefficient, or overly manual.

Cosmos is not just large—it's structurally diverse and dynamically sequenced. Efficient execution isn't just about compute throughput. It's about system-level adaptability. And that's exactly what Rebellions set out to build.



[Figure 1. Cosmos-Predict1-7B + Cosmos Guardrail diagram]

A Paradigm Shift: Enabling the Unexpected

Rebellions is the first to run Cosmos-Predict1-7B in real time on a commercial NPU—not by brute-forcing performance, but by building a stack that is inherently flexible and scalable. This wasn't a one-off integration.

The Anatomy of Innovation: Full-Stack Mastery

Cosmos ran on ATOM™ because every layer of the stack–from compiler to runtime to silicon–was built to adapt to architectural diversity and scale with emerging models.

Where others optimize for one model, we enable many. That adaptability isn't bolted on—it's built in. Modern AI models demand infrastructure that evolves with them. GPU-based systems offer performance but often rely on legacy toolchains, fragmented software layers, and vendor-locked environments. Rebellions takes a system-level approach. By developing the full stack in-house—from compiler to runtime to hardware—we've built a cohesive platform designed to absorb architectural diversity, not resist it. Instead of asking models to conform to infrastructure, we shape infrastructure to fit the model. This principle is what made Cosmos on ATOM™ not just possible—but replicable.

[Figure 2. Rebellions Full-stack System Diagram (SW to HW)]

Software Innovations: Adaptability by Design

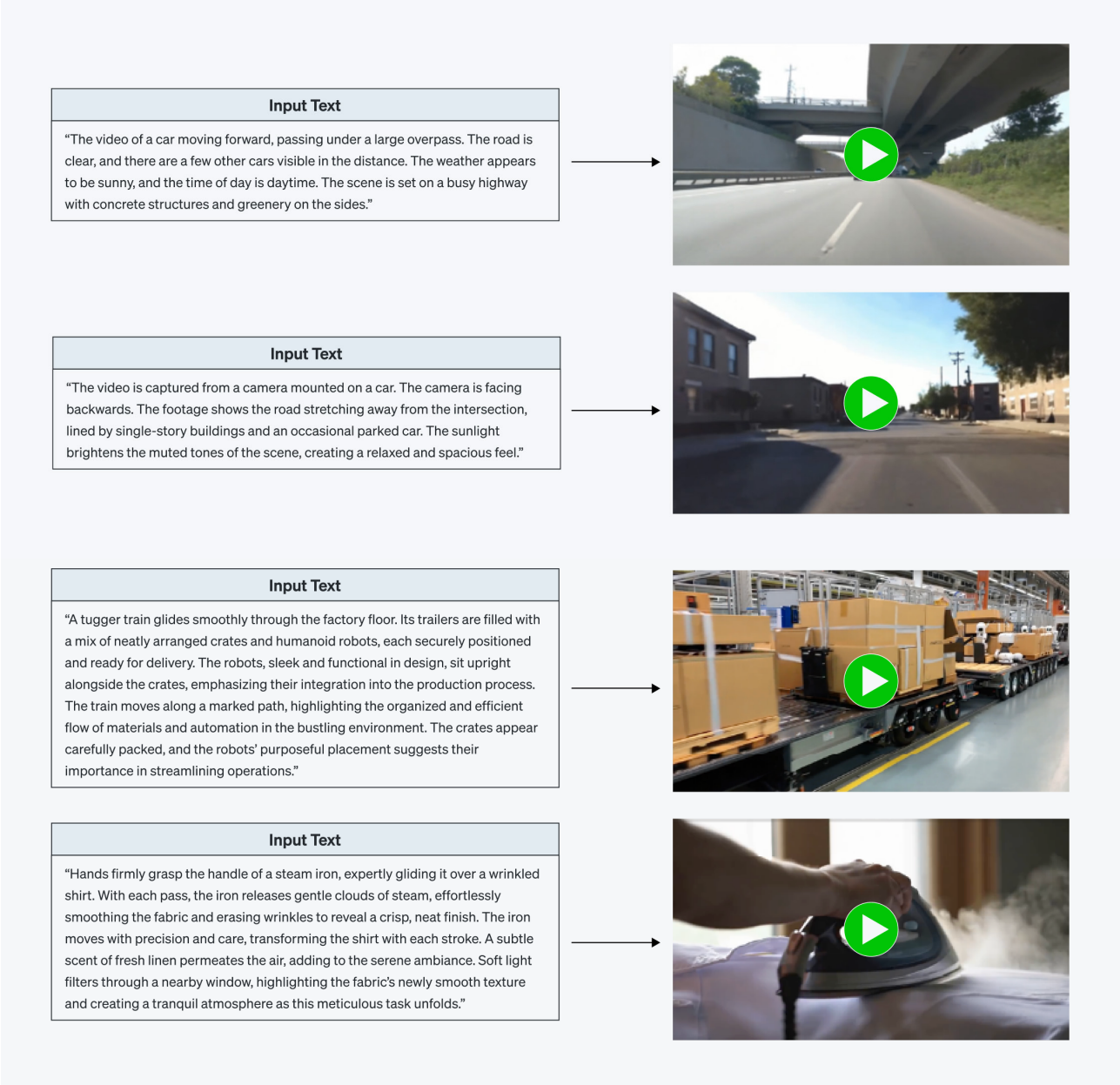
- Modular compiler stack: Adaptability begins with modularity. Rebellions built a compiler stack designed for diversity: with a rich primitive library to support more than 200 distinct models, a front-end compiler that parses transformers, convolutions, and diffusion models, and a back-end that generates hardware-optimized code with minimal hand-tuning.
- Unified runtime orchestration: No two models behave alike—and our runtime treats them accordingly. Rebellions' runtime dynamically orchestrates memory, compute,

and kernel execution flows based on model behavior—maximizing utilization while adapting to layer-wise variation in workloads like Cosmos. It also incorporates predictive DMA scheduling to preload data before it's needed—ensuring that inputs arrive just-in time and compute stalls are avoided even under asynchronous, multi-stage workloads.

- Scalable communication infrastructure: Scalability isn't optional—it's essential. Our "Rebellions Scalable Design" enables multi-card inference through high-bandwidth communication layers and tensor-parallel execution. It's what allows Cosmos to span across NPUs without introducing latency or interconnect bottlenecks.

Hardware Innovations: Purpose-Built for Flexibility

- Flexible compute architecture: The hardware must move with the model. Rebellions engineered ATOM™'s compute cores to support a wide operator set and dynamic execution paths—so that model-level flexibility translates seamlessly into silicon-level efficiency. This flexibility was essential for handling Cosmos' diverse operations without requiring model-specific kernel rewrites.
- Memory-compute coordination: Modern models don't compute linearly—and neither should memory. Rebellions designed this coordination logic to track runtime behavior and adjust compute memory flow in real time—making it inherently capable of handling bursty, irregular phases like those in Cosmos without special case engineering.
- High-bandwidth interconnect for multi-card scaling: Scaling across silicon shouldn't mean starting from scratch. Rebellions developed custom interconnect IP to enable high-throughput communication across NPUs. This ensures low-latency coordination for tensor-parallel workloads like Cosmos without introducing bottlenecks at scale.



[Figure 3. From Prompt to Video: Cosmos-Predict1-7B Inference on ATOM™]

Watch Inference Results

Conclusion: Pioneering a New Era of Accessible AI

Cosmos on ATOM™ is more than a milestone. It's a validation of an architectural philosophy: systems must adapt to models, not the other way around. Rebellions didn't build a chip for one model. We built a platform for all models still to come. In a world of exploding model complexity and compute asymmetry, the future belongs to systems that can adapt fast, scale cleanly, and execute precisely. That's what Rebellions is building. Cosmos is just the beginning.