The Multimodal Mind: Perceive, Imagine, Act

Siddharth Choudhary

***Abstract*—The next frontier in AI is a single model that understands the physical world well enough to imagine its futures and act to shape them. The path runs through a natively multimodal foundation model—one that perceives, reasons, and generates across all modalities (image, video, text, speech, action)—organized around a three-stage cognitive loop: Perceive → Imagine → Act. Biological intelligence follows that loop without thinking about it, yet current AI breaks it into disconnected pipelines, each trained by a different team on a different dataset. This essay argues for building it as one model: a mixture-of-transformers backbone that runs the loop, a staged pretraining curriculum that climbs from understanding to generation to action, and a self-play flywheel that grounds learning in physics. The unifying observation is that every major AI capability challenge—controllable video, computer use, robotics, world simulation, autonomous driving—is a special case of multimodal conditional generation, so one architecture subsumes them all.**[1]

I. The Thesis

The next frontier in AI is a single model that understands the physical world well enough to *imagine* its futures and *act* to shape them. The path runs through a natively multimodal foundation model—one that perceives, reasons, and generates across all modalities (image, video, text, speech, action)—organized around a three-stage cognitive loop: Perceive → Imagine → Act.

Biological intelligence follows that loop without thinking about it: observe the world, mentally simulate what should happen, then intervene. Current AI breaks the loop into disconnected pipelines—a perception model here, a planner there, a controller bolted on the end, each trained by a different team on a different dataset. I propose building it as one model.

Every major AI capability challenge today—controllable video, computer use, robotics, world simulation, autonomous driving, creative tools—is a special case of *multimodal conditional generation*. One architecture subsumes them all [19]. Improvements to the shared backbone benefit every task at once.

The rest of this paper makes that concrete: the cognitive loop, the mixture-of-transformers architecture that runs it, the staged pretraining recipe, the self-play flywheel, and the unification of today's siloed problems.

II. Cognitive Loop: Perceive, Imagine, Act

Three stages, run in a cycle. Each is inference in the *same* unified multimodal space—not three separate modules wired together.

1) **Perceive.** The model ingests the current state across all modalities—video, audio, proprioception, language—into a unified representation. Perception is inference, not a preprocessing step.
2) **Imagine.** Before acting, the model thinks in multimodal space. Not just text chain-of-thought, but a *mental video* of the planned trajectory: predicted proprioception, anticipated contact forces. This multimodal imagination *is* the world model [11], [12].
3) **Act.** The model generates whatever output the task requires—motor commands, keyboard/mouse, speech—conditioned on the imagined plan, not on reactive perception alone. The action changes the world, new percepts arrive, and the loop repeats.

Consider the canonical example: a model deciding to pick up a glass. In the *Perceive* stage, video, audio, proprioception, and language stream in and are inferred into one unified representation; the model sees the glass, the table, and its own hand position, all in the same space. In the *Imagine* stage, the model rolls out a mental video of the plan: the hand approaching, fingers closing, the predicted weight shift and contact forces. In the *Act* stage, conditioned on the imagined plan, the model emits motor commands that close the grasp; the world changes, fresh percepts arrive, and the loop returns to Perceive.

The key idea: **Imagine and Act use the same generative machinery as Perceive**. Imagining "the hand approaching, fingers closing, the weight shifting" is video + proprioception generation. Acting is action generation. They are not bolted-on heads—they are the same model generating in different output modalities. That is what the next section's architecture buys us.

III. One Architecture: Mixture of Transformers

If one model has to perceive *and* generate images, video, audio, and actions, a single dense network creates a problem: modalities interfere. The statistics of a video frame and a motor command have nothing in common, and forcing the same feed-forward weights to model both means each capability dilutes the other.

Start with how the model even sees a video frame or a motor command: **everything becomes tokens**. A per-modality encoder maps each input into one shared token space—text by byte-pair encoding, images and video into VAE latent patches, audio into neural-codec tokens, actions into discretized or continuous control tokens. The transformer then sees a single interleaved stream, each token tagged with its modality.

The architecture I use is a **mixture of transformers (MoT)** [1]. The trick is to split the network by *modality*:

[1] https://itzsid.github.io/publications/multimodal-mind.html

- **Every weight is modality-specific.** Each modality gets its own copy of the entire transformer block—the query/key/value and output projections, the feed-forward network, and the RMSNorm layers. Those weights *are* the experts. A token's modality deterministically selects which expert processes it; there is no learned router.
- **Only the attention *operation* is global.** Each token's Q, K, and V are produced by its own modality's projections, but attention then runs over the combined sequence—every token reads all prior context regardless of modality, so a later video token attends to earlier text and action tokens. The pattern tracks the objective: autoregressive tokens attend *causally* to the past, while a block of tokens being generated by diffusion attends *bidirectionally* within itself, conditioned on that causal context [2], [3]. The operation has no weights of its own; it is the wire that makes this *one* model, while every learnable parameter stays modality-specific.

The experts share no weights—only a single representational space they read and write through global attention; that shared space is the *unified multimodal space* the loop runs in. Concretely, for token i with modality m_i, every projection carries the modality subscript while attention mixes across the whole sequence:

$$\begin{aligned} q_i &= W^{Q}_{m_i} x_i, \quad k_i = W^{K}_{m_i} x_i, \quad v_i = W^{V}_{m_i} x_i \\ \alpha_{ij} &= \mathrm{softmax}_j\big(q_i^\top k_j / \sqrt{d}\big) \\ y_i &= \mathrm{FFN}_{m_i}\Big(\textstyle\sum_j \alpha_{ij}\, v_j\Big) \end{aligned} \tag{1}$$

Every learnable matrix carries the subscript m_i. The attention coefficients α_{ij} are activations, not parameters—computed at runtime—and the mixing they perform runs globally across modalities, which is exactly what lets one modality's tokens read another's.

The payoff shows up in the next section: because every weight a token touches lives in its modality's expert, you can **freeze an entire expert**—its attention projections included—while training another, and global attention still lets the new expert read everything the frozen ones know.

Notice that the attention *operation* never changes—only which modality experts (and therefore which Q/K/V, output, FFN, and norm weights) are present and active. Drop video and audio and you have a vision-language understanding model. Add the action expert and the same backbone becomes an agent. **One model, reconfigured by which experts are switched on.**

IV. The Pretraining Recipe: A Staged Curriculum

The recipe mirrors how we train LLMs, but extends it into multimodal space [19]. Understanding is autoregressive next-token prediction; generation uses the objective each modality calls for—diffusion for continuous signals like image, video, audio, and continuous motor control [2]—so the same backbone is autoregressive where it perceives and diffusion where it generates. Summed over the understanding token positions $\mathcal{U}$ and the generation positions $\mathcal{G}$, one pretraining loss carries both objectives:

$$\mathcal{L} = \underbrace{-\sum_{i\in\mathcal{U}} \log p_\theta\big(x_i \mid x_{<i}\big)}_{\text{autoregressive — understand}} + \lambda \underbrace{\sum_{i\in\mathcal{G}} \mathbb{E}_{t,\epsilon}\big\| \epsilon - \epsilon_\theta\big(x_i^{(t)},\, t \mid c_i\big)\big\|^2}_{\text{diffusion — generate}} \tag{2}$$

where c_i is the multimodal context attended to by token i, and λ balances the two objectives, which live on different scales. You do not learn to perceive, imagine, and act all at once: generation is harder than understanding, and each generative capability builds on the one before it. So pretraining climbs a ladder:

MM understanding → image gen. → video / audio gen. → action gen.

Two rules govern the climb:

- **Add one expert per stage.** Each new stage introduces a new MoT generation expert (Section III) on top of everything learned so far.
- **Freeze everything earlier.** When training a later generation component, the earlier components are frozen. Only the newest expert learns.

Formally, at stage s only the new expert's parameters θ_s are trained, with every earlier expert held fixed:

$$\theta_s^\star = \arg\min_{\theta_s}\ \mathcal{L}_s\big(\theta_s\,;\, \underbrace{\theta_1^\star, \ldots, \theta_{s-1}^\star}_{\text{frozen}}\big) \tag{3}$$

A. Why this order?

The sequence is not arbitrary—each arrow is a dependency. *Understanding before generation*: you cannot imagine a coherent future until you can perceive the present; stage 1 builds the perceptual prior that everything conditions on. *Image before video*: a video is images over time, so the video expert reuses the frozen image expert's spatial priors and only has to learn temporal dynamics—a much smaller lift than learning pixels and motion together. *Generation before action*: action is conditioned on *imagination* (Section II), so the model must be able to roll out a predicted video/proprioception trajectory before it can choose a motor command that achieves it; the action expert comes last, reading from a frozen world model [23].

B. The data pyramid

Staging is also the only practical option, because **data gets scarcer at every stage**. The corpus of text, images, video, and audio for *understanding* is internet-scale. Paired data for high-quality image *generation* is smaller; video and audio generation smaller still; and action data—robot trajectories, labeled screen recordings, driving logs—is the scarcest of all. Each stage trains on roughly an order of magnitude less data than the one before. So we spend the abundant data first to build the strongest possible backbone, then climb into the data-poor regimes on top of it.

C. Why freeze?

Freezing earlier components does three things at once. *No catastrophic forgetting*: hard-won perception and image priors cannot be eroded by the noisier gradients of a new, harder generative task; this matters most because of the data pyramid, since if everything trained jointly the tiny action dataset would be swamped by—and would perturb—the enormous understanding corpus. *Clean credit assignment*: the newest expert is the only thing learning, so every gradient is attributable to it, and nothing else can drift to "explain away" its errors. *Cheaper training*: a frozen expert is just a forward pass—no optimizer state, no backward pass through its weights, so each later stage is far cheaper than training the whole stack jointly. Because every weight—attention projections included—lives inside a modality expert, freezing an expert freezes its Q/K/V too. The global attention operation has no parameters of its own; it simply lets each newly added expert read and write through the frozen experts' representations.

D. The learning-rate schedule

All of this rides on a single learning-rate schedule with three phases: a short **warm-up** that ramps the LR to its peak, a long **constant-LR** (CLR) plateau that does the bulk of the work, and a **ramp-down** that anneals the LR toward zero. The staged curriculum maps directly onto it. The first **50% of the CLR plateau is understanding only**—the data-rich foundation—and the generation stages (image, then video/audio, then action) are layered into the second half. The ramp-down is reserved for **high-quality data across every modality**, polishing the whole stack as the LR decays to zero.

The staged-and-frozen recipe turns one terrifying joint-training problem into four tractable ones. Each stage asks a single, well-posed question—*given everything I already know, how do I generate this one new modality?*—and answers it without disturbing the rest.

V. Post-training and the Physics Flywheel

Pretraining learns the prior over how the multimodal world works. Two stages turn that prior into a capable agent:

- **Post-training (SFT).** Supervised fine-tuning on curated Perceive $\rightarrow$ Imagine $\rightarrow$ Act demonstrations, with explicit multimodal reasoning traces—the imagination made legible.
- **RL via self-play in simulation** with automatic reward signals. This is where the model becomes *self-improving*.

In simulation, the model runs its full loop and grades itself automatically on three questions. *Did what I imagined actually happen?*—imagination accuracy. *Did I succeed?*—task completion. *Does my prediction violate physics?*—consistency, checked against engine constraints. Call it *RL from physics feedback* (RLPF).

Self-play maximizes expected return under a composite reward that folds these together:

$$\max_{\theta} \; \mathbb{E}_{\tau\sim\pi_\theta}\Big[\textstyle\sum_t R_t\Big], \qquad R = \alpha\, R_{\text{task}} + \beta\, R_{\text{imag}} + \gamma\, R_{\text{phys}} \tag{4}$$

with R_{imag} the imagination accuracy and R_{phys} the physics-consistency term (RLPF).

Trace one lap to see why it *compounds*. A better world model makes the model's imagination more accurate, so it can plan against a mental simulation it trusts—better plans. Better plans yield better actions, which succeed more often. Those successful rollouts ("this plan led to this outcome") are exactly the labeled examples you want, so they become richer training data—which trains an even better world model. Each lap raises the floor for the next.

A. Why physics, not just text rewards?

Text models can self-improve this way too—but only where an automatic grader exists. Math has checkable answers, code has unit tests, and RL from verifiable rewards has driven much of the recent progress in exactly those domains [29]. Two things limit it. First, **breadth**: verifiable graders cover a narrow slice of what we care about, and everything else falls back on learned proxies—reward models, LLM judges—which are themselves models that can be reward-hacked and carry no ground truth about the physical world. Second, **what the signal improves**: a unit test sharpens a policy, but it does not teach the model how the world works. Physics fixes both. It is a broad, grounded grader across the entire embodied space—did the glass actually lift? did the motion obey gravity?—and that signal flows back into the world model itself. Physics is to physical reality what unit tests are to code.

The catch is that the whole loop runs *inside* a simulation, so its ceiling is **simulation fidelity**. Train against a simulator that gets physics wrong and the model learns from a lie—the rewards are noise and nothing compounds. And neither obvious option is enough on its own: a hand-built physics engine is accurate but cannot render the visual richness of the real world, while real video is realistic but you cannot act inside it. The essential ingredient is therefore a *learned* world simulator—trained on real video for realism and interactivity [13], and corrected by physics engines for physical consistency [20]. Low fidelity and the flywheel slips; high fidelity and each cycle multiplies the last. The whole strategy lives or dies on the quality of the imagined world.

VI. One Model, Many Problems

Here is the payoff of building it as one model. Every major AI capability challenge is a different *route* through the same architecture—a particular choice of input modalities and output modality. Familiar combinations name an established task; the rest are potential applications—routes the same model could serve, with no household name yet.

Each is the same conditional distribution—choose which modality streams are given and which are generated:

$$y \sim p_\theta\big(y \mid x_1, x_2, \ldots, x_k\big) \tag{5}$$

Table I reads off several familiar tasks as routes through this one conditional model. Today each of these is a separate model, dataset, and research community. The Perceive $\rightarrow$ Imagine $\rightarrow$ Act framework unifies them as different routes

TABLE I
TODAY'S SILOED PROBLEMS AS ROUTES THROUGH ONE MULTIMODAL CONDITIONAL GENERATOR: A CHOICE OF INPUT MODALITIES AND OUTPUT MODALITY.

inputs (perceive)	output (act)	task
text	image	text-to-image
text	video	text-to-video
text, image	video	image animation
image, action	video	world simulation
video, language	action	robot control (VLA)
screen, instruction	action	computer use
video, sensors	action	autonomous driving
audio, text	speech	speech assistant

through one architecture—and improvements to the shared backbone benefit every task simultaneously. That is the whole bet: **build the multimodal mind once, and every capability comes along for the ride.**

VII. RELATED WORK

This essay synthesizes several lines of work. The architecture draws on sparse, modality-decoupled transformers and unified autoregressive-plus-diffusion models [1], [2], [3], [4], [5], [6], [7], [8], [9], [10]; the cognitive loop and physics flywheel build on the world-models tradition [11], [12], [13], [14], [15], [16], [17], [18]; and the physical-AI framing tracks a recent wave of world foundation models [19], [20], [21], [22], [23], [24], [25], [26], [27], [28]. Some of it—most directly NVIDIA's Cosmos 3 [19]—arrived at strikingly similar conclusions: a two-tower mixture-of-transformers that unifies understanding, world generation, and action for physical AI. The self-play training loop draws on reinforcement learning from verifiable rewards and modern post-training [29], [30], [31], [32], [33], [34], and on the self-play-and-planning lineage [35], [36].

REFERENCES

[1] W. Liang et al., "Mixture-of-Transformers: A Sparse and Scalable Architecture for Multi-Modal Foundation Models," TMLR, 2025. arXiv:2411.04996.

[2] C. Zhou et al., "Transfusion: Predict the Next Token and Diffuse Images with One Multi-Modal Model," 2024. arXiv:2408.11039.

[3] Chameleon Team (Meta FAIR), "Chameleon: Mixed-Modal Early-Fusion Foundation Models," 2024. arXiv:2405.09818.

[4] J. Xie et al., "Show-o: One Single Transformer to Unify Multimodal Understanding and Generation," ICLR, 2025. arXiv:2408.12528.

[5] X. Wang et al. (BAAI), "Emu3: Next-Token Prediction is All You Need," 2024. arXiv:2409.18869.

[6] C. Wu et al. (DeepSeek-AI), "Janus: Decoupling Visual Encoding for Unified Multimodal Understanding and Generation," 2024. arXiv:2410.13848.

[7] X. Chen et al. (DeepSeek-AI), "Janus-Pro: Unified Multimodal Understanding and Generation with Data and Model Scaling," 2025. arXiv:2501.17811.

[8] J. Lu et al. (AI2), "Unified-IO 2: Scaling Autoregressive Multimodal Models with Vision, Language, Audio, and Action," CVPR, 2024. arXiv:2312.17172.

[9] T. Li et al., "Autoregressive Image Generation without Vector Quantization (MAR)," NeurIPS, 2024. arXiv:2406.11838.

[10] C. Deng et al. (ByteDance Seed), "Emerging Properties in Unified Multimodal Pretraining (BAGEL)," 2025. arXiv:2505.14683.

[11] D. Ha and J. Schmidhuber, "World Models," NeurIPS, 2018. arXiv:1803.10122.

[12] D. Hafner et al., "Mastering Diverse Domains through World Models (DreamerV3)," Nature, 2025. arXiv:2301.04104.

[13] J. Bruce et al., "Genie: Generative Interactive Environments," ICML, 2024. arXiv:2402.15391.

[14] V. Micheli et al., "Transformers are Sample-Efficient World Models (IRIS)," ICLR, 2023. arXiv:2209.00588.

[15] E. Alonso et al., "Diffusion for World Modeling: Visual Details Matter in Atari (DIAMOND)," NeurIPS, 2024. arXiv:2405.12399.

[16] A. Hu et al. (Wayve), "GAIA-1: A Generative World Model for Autonomous Driving," 2023. arXiv:2309.17080.

[17] S. Gao et al., "Vista: A Generalizable Driving World Model with High Fidelity and Versatile Controllability," NeurIPS, 2024. arXiv:2405.17398.

[18] A. Bar et al. (Meta FAIR / NYU), "Navigation World Models," CVPR, 2025. arXiv:2412.03572.

[19] NVIDIA, "Cosmos 3: Omnimodal World Models for Physical AI," 2026. arXiv:2606.02800.

[20] NVIDIA, "Cosmos World Foundation Model Platform for Physical AI," 2025. arXiv:2501.03575.

[21] NVIDIA, "World Simulation with Video Foundation Models for Physical AI (Cosmos-Predict2.5)," 2025. arXiv:2511.00062.

[22] NVIDIA, "Cosmos-Reason1: From Physical Common Sense to Embodied Reasoning," 2025. arXiv:2503.15558.

[23] NVIDIA, "GR00T N1: An Open Foundation Model for Generalist Humanoid Robots," 2025. arXiv:2503.14734.

[24] Google Robotics, "RT-1: Robotics Transformer for Real-World Control at Scale," 2022. arXiv:2212.06817.

[25] Google DeepMind, "RT-2: Vision-Language-Action Models Transfer Web Knowledge to Robotic Control," CoRL, 2023. arXiv:2307.15818.

[26] M. J. Kim et al. (Stanford), "OpenVLA: An Open-Source Vision-Language-Action Model," CoRL, 2024. arXiv:2406.09246.

[27] K. Black et al. (Physical Intelligence), "π_0: A Vision-Language-Action Flow Model for General Robot Control," 2024. arXiv:2410.24164.

[28] Octo Model Team, "Octo: An Open-Source Generalist Robot Policy," RSS, 2024. arXiv:2405.12213.

[29] DeepSeek-AI, "DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning," Nature, 2025. arXiv:2501.12948.

[30] L. Ouyang et al. (OpenAI), "Training Language Models to Follow Instructions with Human Feedback (InstructGPT)," NeurIPS, 2022. arXiv:2203.02155.

[31] H. Lightman et al. (OpenAI), "Let's Verify Step by Step," ICLR, 2024. arXiv:2305.20050.

[32] Z. Shao et al. (DeepSeek-AI), "DeepSeekMath: Pushing the Limits of Mathematical Reasoning (GRPO)," 2024. arXiv:2402.03300.

[33] N. Lambert et al. (AI2), "Tülu 3: Pushing Frontiers in Open Language Model Post-Training," 2024. arXiv:2411.15124.

[34] Kimi Team (Moonshot AI), "Kimi k1.5: Scaling Reinforcement Learning with LLMs," 2025. arXiv:2501.12599.

[35] D. Silver et al. (DeepMind), "Mastering Chess and Shogi by Self-Play with a General Reinforcement Learning Algorithm (AlphaZero)," Science, 2018. arXiv:1712.01815.

[36] J. Schrittwieser et al. (DeepMind), "Mastering Atari, Go, Chess and Shogi by Planning with a Learned Model (MuZero)," Nature, 2020. arXiv:1911.08265.