

Does multimodality still affect the latest humanoids

Notes from a chat with Claude, October 3–4, 2026.

Question: Does the averaging (multimodality) problem happen with the latest humanoids?

Yes, but mostly in a milder form. The crash-into-the-middle version has largely been designed out of today's leading humanoid systems. What remains is the robot changing its mind between options, which engineers still actively work on.

How current humanoids avoid averaging

- **Generative action heads.**
 - Boston Dynamics' Atlas behavior model is a 450M-parameter diffusion transformer trained with flow matching. It predicts 1.6-second action chunks at 30 Hz, commanding hands, feet and torso together.
 - NVIDIA's GR00T N1.7 and Physical Intelligence's $\pi 0.5$ also use flow matching.
 - All three sample one option instead of blending options.
- **Imagine first, then act.** 1X's NEO uses a video model to imagine the next moments, then a second model works out the motor commands that produce that imagined video. Once the video shows the robot going left, the second model has nothing ambiguous to average.
- **Learned balance controllers.** These are trained in simulation, by trial and reward, to follow a target motion handed down from the layer above. NVIDIA's SONIC is one. They aren't copying conflicting demos, so there's nothing to average.
- **Figure is the interesting exception.** In Figure's published description (Feb 2025), Helix is trained end-to-end with "a standard regression loss", which is the averaging kind:
 - A large slow model (7–9 Hz) passes a summary vector to a small fast model (80M parameters, 200 Hz) that outputs continuous motor commands.
 - Figure hasn't said how it copes with the problem. Two likely reasons:
 - Its operators collect consistent demos.
 - At 200 Hz, with the robot sensing its own motion, once it drifts even slightly left, the demos from that state all say "keep going left." The tie breaks within a fraction of a second.
 - Figure hasn't published the training loss for its newer versions, and Tesla hasn't published Optimus's policy design at all.

Where it still shows up

- **Changing its mind between chunks.** Physical Intelligence's paper on real-time chunking says "adjacent chunks may jump between different modes (or 'strategies')", which causes jerky motion the model never saw in training. Smoothing by averaging the two chunks "may only make matters worse"; that is the averaging problem coming back. Their fix freezes the actions that will run while the next chunk is being computed, and generates the rest of the new chunk to continue them.
- **Hesitating at decision points.** Where demos split and the output still gets blended, the robot can slow down or hover before committing. That's the regression problem in its milder form.

- **Choosing by popularity, not quality.** A generative policy picks each option about as often as demonstrators did. If 70% went left, it goes left 70% of the time, even where right is better. Reinforcement-learning post-training, such as Physical Intelligence's Recap, shifts the odds toward options that work.
- **More either/or choices for a whole body.** A humanoid also has to choose which hand to use, which foot steps first, and whether to step or lean. A whole-body policy like Atlas's faces more forks than an arm does.

What it means if you fine-tune GR00T N1.7 or $\pi 0.5$

The averaging crash isn't your main risk. Two practical things matter more:

- **Smooth handoffs between chunks:** use real-time chunking or something similar.
- **Consistent demos:** write a collection protocol so operators handle the same situation the same way. Where a choice is legitimate, put it in the instruction ("go around on the left") so the model can see why each demo went the way it did.

Sources:

- [Figure: Helix](#)
- [Boston Dynamics: Large behavior models and Atlas](#)
- [Real-Time Execution of Action Chunking Flow Policies \(arXiv\)](#)
- [How Humanoids Learn](#) (1X, GR00T N1.7 and SONIC details)