> TL;DR OmniAvatar adapts Wan 2.1 into an audio-driven avatar generator with full-body motion. Pixel-wise multi-hierarchical audio embeddings improve lip sync, while LoRA-based training keeps prompt creativity intact. The release ships LoRA weights (14B and 1.3B) plus inference code, so teams can render 480p avatars with controllable prompts and audio guidance on-prem.

What is OmniAvatar?

OmniAvatar is a research framework for producing audio-driven avatar videos that move beyond face-only animation. The team combines Wan 2.1 text-to-video backbones with new audio conditioning so characters maintain lip sync and natural body dynamics, even in conversational or performance settings. The work was posted to arXiv on 23 June 2025 and open-sourced with inference code a day later.

The method introduces a pixel-wise multi-hierarchical audio embedding that slots into the latent space of the diffusion model. By pairing that with lightweight LoRA adaptation, OmniAvatar keeps Wan's ability to follow creative prompts while fusing in speech nuances that drive torso, arm, and facial motion in sync with the soundtrack.

Links:

- Paper: https://arxiv.org/abs/2506.18866
- Project page: https://omni-avatar.github.io/
- GitHub: https://github.com/Omni-Avatar/OmniAvatar
- Hugging Face weights: https://huggingface.co/OmniAvatar/OmniAvatar-14B

Key ideas

- Pixel-wise multi-hierarchical audio embeddings: audio is encoded across scales so the diffusion latent receives fine-grained phoneme cues and broader rhythm, sharpening lip sync in diverse scenes (abstract).
- Adaptive body animation: conditioning extends to upper-body pose and gestures, so avatars react naturally in podcasts, dialogues, dynamic scenes, and singing use cases (abstract + project page).
- LoRA-based alignment: OmniAvatar adds LoRA adapters on top of Wan 2.1 (14B and 1.3B) rather than retraining end-to-end, preserving prompt controllability for styling and camera direction (GitHub README).
- Decoupled guidance: guidance scales let you tune prompt adherence versus audio faithfulness (`guidance_scale` vs `audio_scale`), with audio CFG in the recommended 4–6 band for reliable lip sync (GitHub README).
- Runtime efficiency knobs: supports FSDP, TeaCache, and per-layer persistence settings so teams can trade VRAM for speed—e.g., 4×A800 with FSDP drops sampling to 4.8 s/it while keeping ~14.3 GB per GPU (GitHub README).

Model lineup & availability

- OmniAvatar-14B LoRA + audio condition weights (pairs with Wan2.1-T2V-14B and wav2vec2-base-960h).
- OmniAvatar-1.3B LoRA + audio condition weights for faster runs on smaller hardware.
- Pretrained audio encoder: facebook/wav2vec2-base-960h.
- Example prompts, configs, and video samples ship in the repo (`examples/`, `configs/`).

Weights are hosted on Hugging Face; download via `huggingface-cli` or from the linked demo space.

Quickstart (from repo docs)

Clone and install:

```
git clone https://github.com/Omni-Avatar/OmniAvatar
cd OmniAvatar
pip install torch==2.4.0 torchvision==0.19.0 torchaudio==2.4.0 \
  --index-url https://download.pytorch.org/whl/cu124
pip install -r requirements.txt
# Optional: accelerate attention
pip install flash_attn
```

Grab models (14B example):

```
mkdir -p pretrained_models
pip install "huggingface_hub[cli]"
huggingface-cli download Wan-AI/Wan2.1-T2V-14B --local-dir ./pretrained_models/Wan2.1-T2V-14B
huggingface-cli download facebook/wav2vec2-base-960h --local-dir ./pretrained_models/wav2vec2-base-960h
huggingface-cli download OmniAvatar/OmniAvatar-14B --local-dir ./pretrained_models/OmniAvatar-14B
```

Run inference (480p for now):

```
# 14B
torchrun --standalone --nproc_per_node=1 scripts/inference.py \
  --config configs/inference.yaml \
  --input_file examples/infer_samples.txt

# 1.3B
torchrun --standalone --nproc_per_node=1 scripts/inference.py \
  --config configs/inference_1.3B.yaml \
  --input_file examples/infer_samples.txt
```

Prompting & control tips

- Input format: `[prompt]@@[img_path]@@[audio_path]` per line in `examples/infer_samples.txt`. Leave `img_path` empty for audio+prompt only, or point to a still frame for identity anchoring.
- Guidance knobs: keep `guidance_scale` and `audio_cfg` (or `audio_scale` when split) between 4–6. Bump audio guidance if lip sync lags; raise prompt guidance for stricter style.
- Steps vs. quality: 20–50 denoising steps are recommended. More steps improve fidelity; fewer accelerate iteration.
- Sequence parallel: set `sp_size` to the number of GPUs for multi-GPU inference; combine with `use_fsdp=True` to squeeze VRAM.
- TeaCache: enable `tea_cache_l1_thresh` (0.05–0.15) for cache-accelerated sampling on repeated prompts.

Practical production notes

- Resolution: official release targets 480p. The team reports training on 30 k tokens per clip; longer clips or 720p need further fine-tuning.
- Performance: single A800, 14B model → ~16 s/it at 36 GB VRAM. With FSDP across 4 GPUs the run drops to 4.8 s/it while keeping memory near 14.3 GB per card.
- Memory levers: set `num_persistent_param_in_dit` (e.g., 7 B) to cap per-device activation storage; combine with `use_fsdp=True` for large prompts.
- Audio quality: increase `overlap_frame` from 1 to 13 for smoother body motion (at the cost of error propagation); tighten prompts to cover first frame, behavior, and background.
- Demo access: a Hugging Face Space offers browser inference for quick auditions before running on local GPUs.

References

- arXiv: https://arxiv.org/abs/2506.18866
- Project: https://omni-avatar.github.io/
- GitHub: https://github.com/Omni-Avatar/OmniAvatar
- Hugging Face (14B): https://huggingface.co/OmniAvatar/OmniAvatar-14B
- Hugging Face (1.3B): https://huggingface.co/OmniAvatar/OmniAvatar-1.3B

Notes: Data points reflect the public README and project page as of 24 June 2025. Monitor the repo for updated checkpoints, longer-video support, and new resolution presets.