> TL;DR Hunyuan3D 2 uses a flow-based diffusion transformer for geometry, a dedicated paint model for 4K textures, and a suite of Turbo, mini, and mv checkpoints plus FlashVDM acceleration so teams can turn concept art into production-grade assets on commodity GPUs.

What is Hunyuan3D 2?

Hunyuan3D 2 is Tencent's second generation text and image conditioned 3D asset creator. The stack breaks the problem into shape and texture stages: a large Hunyuan3D-DiT flow transformer builds the mesh, while Hunyuan3D-Paint applies physically based textures that track the input prompt or concept art. The team ships everything under permissive terms alongside a hosted studio, API server, Gradio UI, and Blender add-on.

Links:

- GitHub: https://github.com/Tencent-Hunyuan/Hunyuan3D-2
- Technical report (2.0): https://arxiv.org/abs/2501.12202
- Technical report (2.5): https://arxiv.org/abs/2506.16504
- Demo: https://huggingface.co/spaces/tencent/Hunyuan3D-2
- Official site: https://3d.hunyuan.tencent.com

Why it matters for 3D and marketing teams

- Replace manual sculpt plus texture bake cycles with a prompt-to-ready mesh flow that respects product references.
- Keep iteration velocity high: Turbo and mini checkpoints cut denoising steps while FlashVDM and low VRAM modes keep RTX 4090 builds viable.
- Drop generated props straight into pipelines through GLB, OBJ, or Blender import with consistent UVs and PBR maps.
- Localise launches and campaigns fast by repainting hand-authored meshes with the paint model's texture-only mode.

System architecture highlights

- Two stage pipeline: mesh first, texture second, making it easy to texture imported meshes without regenerating geometry.
- Flow-matching diffusion transformer: scalable DiT backbone that improves alignment between silhouettes and conditioning images.
- High resolution texture synthesis: Hunyuan3D-Paint leans on geometric priors plus diffusion to deliver detailed albedo, normal, and roughness maps.
- Production tooling: official API server, Blender plug-in, and hosted studio reuse the same backend so teams can pick the integration that fits their stack.

Model zoo and accelerators

- Hunyuan3D-2 core: 1.1B DiT for image conditioned shape plus 1.3B paint model for textures.
- Turbo builds: step distillation variants (DiT-v2-0-Turbo, Paint-v2-0-Turbo) reduce sampling steps for faster previews.
- Mini and mv branches: 0.6B mini models for lighter hardware, and multiview (mv) checkpoints when you already have multi-angle renders.
- 2.1 refresh: June 2025 update adds a PBR-aware paint model, VAE encoder, and full training code for teams that need fine-tuning control.
- 2.5 report: June 2025 paper documents detail upgrades and higher fidelity evaluation, signalling ongoing improvements.
- FlashVDM integration: optional enable flag accelerates Turbo pipelines with latest diffusion acceleration research.

It takes about 6 GB of VRAM to generate shapes and roughly 16 GB to add textures end to end.

Benchmark snapshot

Tencent reports Hunyuan3D 2 beating both open and closed baselines across CMMD, FID, FID_CLIP, and CLIP score. Example metrics:

- CMMD: 3.193 versus 3.218 to 3.600 for prior art.

- FID_CLIP: 49.165 compared to 49.744 to 55.866.
- FID: 282.429, a sizable drop from 289 to 306.
- CLIP-score: 0.809, topping peers' 0.779 to 0.806 band.

These gains show up in tighter prompt adherence and richer surface detail in end to end renders.

Running Hunyuan3D 2 locally

Install dependencies (PyTorch version depends on your CUDA build):

```
pip install -r requirements.txt
pip install -e .
# texture modules
cd hy3dgen/texgen/custom_rasterizer
python setup.py install
cd ../../..
cd hy3dgen/texgen/differentiable_renderer
python setup.py install
```

Generate a textured mesh from a reference image:

```
from hy3dgen.shapegen import Hunyuan3DDiTFlowMatchingPipeline
from hy3dgen.texgen import Hunyuan3DPaintPipeline

shape_pipe = Hunyuan3DDiTFlowMatchingPipeline.from_pretrained("tencent/Hunyuan3D-2")
mesh = shape_pipe(image="assets/demo.png")[0]

paint_pipe = Hunyuan3DPaintPipeline.from_pretrained("tencent/Hunyuan3D-2")
textured_mesh = paint_pipe(mesh, image="assets/demo.png")
textured_mesh.export("demo.glb")
```

Spin up a Gradio UI (low VRAM version):

```
python gradio_app.py \
  --model_path tencent/Hunyuan3D-2 \
  --subfolder hunyuan3d-dit-v2-0 \
  --texgen_model_path tencent/Hunyuan3D-2 \
  --low_vram_mode
```

Expose an API for pipeline integrations:

```
python api_server.py --host 0.0.0.0 --port 8080
# simple image to mesh request
data=$(base64 -i assets/demo.png)
curl -s -X POST http://localhost:8080/generate \
```

```
-H "Content-Type: application/json" \
-d "{\"image\": \"$data\"}" \
-o sample.glb
```

Deployment notes and roadmap

- Low VRAM flags disable persistent DiT parameters and lean on FlashVDM for speed without quality collapse.
- Turbo and mini checkpoints help design teams iterate fast, while full-size 2.0 and 2.1 models deliver hero asset fidelity.
- Texture-only workflows let artists repaint handcrafted meshes using the same diffusion priors, which keeps asset libraries fresh without remodelling.
- Official roadmap items include a TensorRT export along with continued releases (HunyuanWorld, RomanTex, MaterialMVP) that plug into the same ecosystem.
- Community wrappers cover Windows bundles, ComfyUI nodes, and Kaggle notebooks, making training labs and agencies easier to bootstrap.

Questions about integrating Hunyuan3D 2 into your asset pipeline or creative ops?

References

- Hunyuan3D-2 (GitHub)
- Hunyuan3D 2.5 (arXiv)
- Stable Diffusion (GitHub)
- Latent Diffusion Models (arXiv)