Open-Source Real-Time Avatar System

Technical Proposal

1. Executive Summary

Production-grade open-source system converting audio to high-fidelity talking avatar video at 30+ FPS. Hybrid 2D/3D rendering achieves photorealism with temporal stability, addressing jitter and artifacts in existing solutions.

Performance: 25-33ms/frame | 4-6GB VRAM | SSIM >0.90 | LSE-C >8.0

2. System Architecture

Pipeline

Audio → Feature Extraction (2ms) → Motion Generation (6ms) → 3D Render (4ms) → Neural Enhancement (12ms) → Stabilization (2ms) → Output

Audio Features: Wav2Vec2 (MIT) frozen → 768-dim embeddings at 50Hz, FP16 quantized

Motion Network: GRU + Attention (15-30M params) → 68-478 landmarks + blend shapes + head pose, TensorRT FP16

3D Rendering: FLAME model (CC BY 4.0, 5023 vertices) → OpenGL/Vulkan at 512x512

Neural Enhancement: U-Net (12-20M params) → Photorealistic texture, TensorRT FP16

Stabilization: Kalman filter + optical flow + EMA (α=0.75)

Why Hybrid 2D/3D?

3D = geometric stability | 2D neural = photorealism | Pure 2D = jitter | Pure 3D = uncanny | NeRF = too slow

3. Key Design Decisions

Audio: Pretrained Wav2Vec2 (no training needed, robust features)

Motion: Bidirectional GRU, 1.5s context, 200ms lookahead for smoothness

Face Model: FLAME (parametric control) or MediaPipe (Apache 2.0 alternative)

Temporal Consistency: Recurrent arch + temporal attention + Kalman + optical flow + EMA

Training: 50-100hrs diverse speakers | Losses: landmark L2 + perceptual (VGG) + temporal smoothness + adversarial

4. Performance Optimization

Technique	Speedup	Implementation
FP16 quantization	2x	All neural networks
TensorRT	2-4x	Graph optimization, kernel fusion
Depthwise conv	8-9x params	MobileNet-style blocks
Pruning	30-40% smaller	Remove low-magnitude weights
Pipeline async	1.5x	Overlap stages, double buffer

Hardware Performance:

GPU	Sessions	FPS	VRAM/Session
A100 40GB	6-8	35-40	5GB
V100 32GB	4-6	25-30	5GB
T4 16GB	2-3	20-25	4GB

5. Quality Assurance

Metrics: LD <2.5px | LSE-C >8.0 | SSIM >0.90 | LPIPS <0.15 | Jitter <1.0px | MOS >4.0

Artifact Solutions:

- Jitter → Kalman + temporal attention + EMA
- Sync drift → Explicit sync loss, 1.5s context
- Blurring → Perceptual loss, super-resolution
- Rigid expression → Diverse data, micro-expressions, procedural noise
- Mouth artifacts → Separate interior model, mask compositing

Expression Richness: Macro (emotions) + Micro (eyebrows, nostrils) + Eyes (saccades, blinks) + Head (nodding)

Visemes: 20-viseme mapping, co-articulation, context-aware transitions

6. Deployment

VRAM: Audio 0.5GB | Motion 0.8GB | Render 0.3GB | Neural 1.2GB | Buffers 0.7GB | Overhead 0.5GB = 4-5GB

Cloud Costs (30 FPS):

Provider	GPU	$/min
Lambda Labs	A100	$0.011
Vast.ai	A100	$0.007-0.013
AWS	A100	$0.034
GCP	V100	$0.021

Scaling:

- 1-10 sessions: Single A100, $250/mo
- 10-100: Auto-scaling 4-6 A100s, Kubernetes, $1,500-2,000/mo
- 100+: Multi-region, 20+ A100s, $10,000+/mo

Monitoring: Prometheus + Grafana | Latency P95 <40ms | Success rate >99.5% | GPU util 60-80%

7. Open-Source Compliance

All Components Commercial-Friendly:

Wav2Vec2 (MIT) | PyTorch (BSD) | FLAME (CC BY 4.0) | MediaPipe (Apache 2.0) | OpenCV (Apache 2.0) | PyTorch3D (BSD) | Real-ESRGAN (BSD) | TensorRT (NVIDIA EULA) | ONNX Runtime (MIT) | FFmpeg (LGPL)

System License: Apache 2.0

Deliverables: Source code | Pretrained weights (HuggingFace) | Docker image | Training/inference scripts | Deployment guides | Benchmarks

Setup:

```
docker run --gpus all avatar-system:latest python inference.py --audio in.wav --output out.mp4
```

Fully Open Alternative: Replace TensorRT with ONNX Runtime (30% slower but MIT licensed)