

Web3D 2026

13-15 October 2026
Doha, Qatar





BONSAI

Browser-Optimized Neural Gaussian Avatar Streaming

Sina Kitz, Simon Katz, Jan Spindler, Zhastay Yeltay,
Reza Rastegari, Julian Kaltheuner, Patrick Stotko, Reinhard Klein

University of Bonn

Web3D 2026

High-quality Neural Gaussian Avatars on the Web?

UNIVERSITÄT BONN



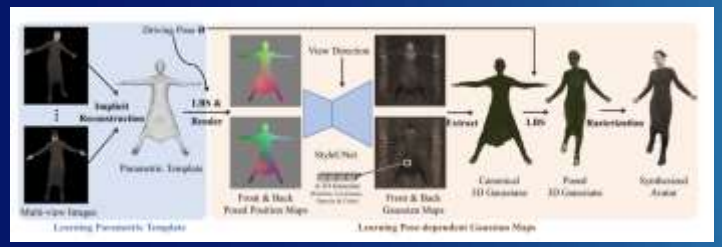
High visual fidelity
Pose-dependent neural decoding

~100k splats
Decoder evaluated every frame

Browser constraints:
compute • memory traffic • delivery

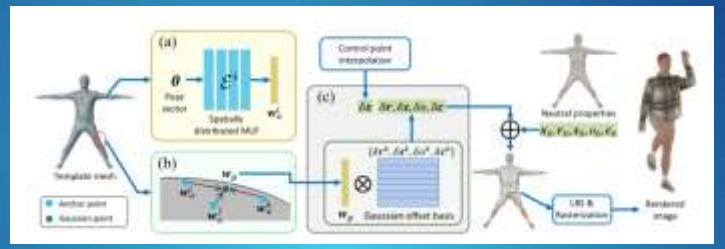
The rendering primitive IS efficient.
The neural avatar around it is NOT automatically Web-ready.

Related High-Fidelity Gaussian Avatars



Animatable Gaussians: Learning Pose-dependent Gaussian Maps for High-fidelity Human Avatar Modeling (Li et al. 2024)

Pose-dependent Gaussian avatars using UV-space neural features.



Real-time High-fidelity Gaussian Human Avatars with Position-based Interpolation of Spatially Distributed MLPs (Zhang et al. 2025)

Compact per-Gaussian MLPs predict pose-dependent Gaussian attributes.



PiG-Avatar: Hierarchical Neural-Field-Guided Gaussian Avatars (Kaltheuner et al. 2026)

Self-organizing anchors with progressive LoDs.

Related Efficient Gaussian Avatars



TaoAvatar: Real-Time Lifelike Full-Body Talking Avatars for Augmented Reality via 3D Gaussian Splatting (Chen et al. 2025)

Compact real-time Gaussian talking-avatar representation



SqueezeMe: Mobile-Ready Distillation of Gaussian Full-Body Avatars (Iandola et al. 2025)

Mobile-ready Gaussian full-body avatars using a compact linear representation



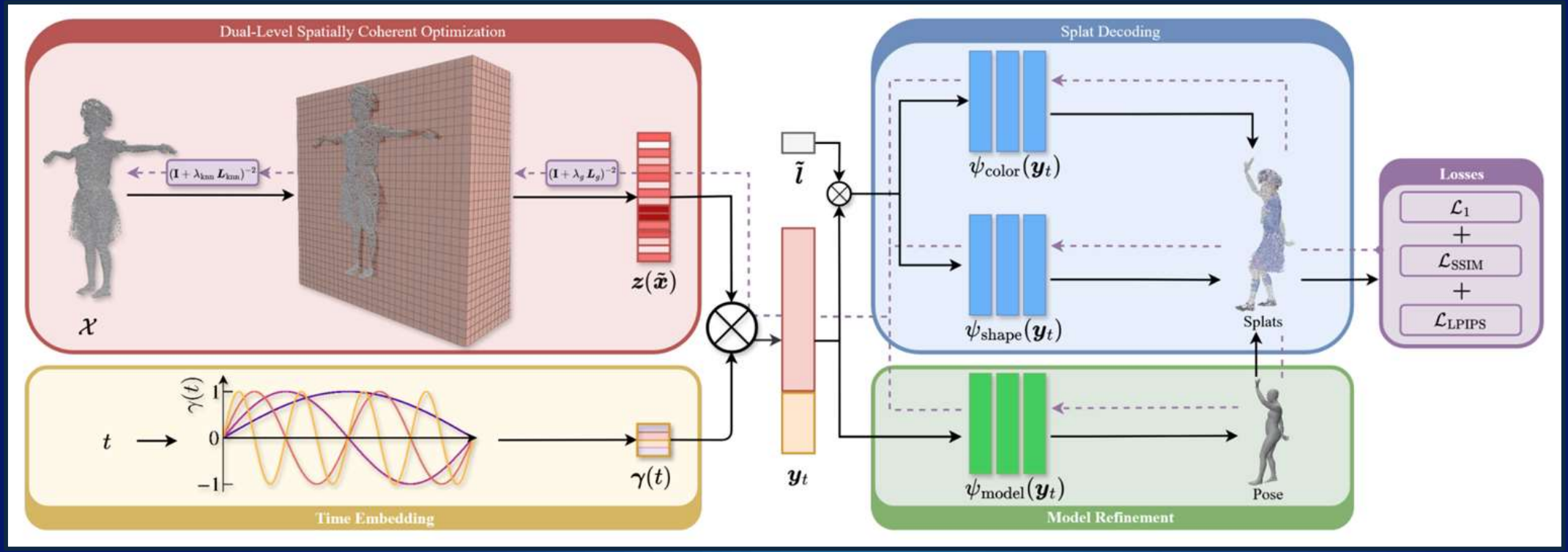
High-Fidelity Mobile Avatars with Pruned Local Blendshapes (Iandola et al. 2026)

Replaces expensive neural decoding with local linear blendshapes

Existing approaches
Design / train a compact representation

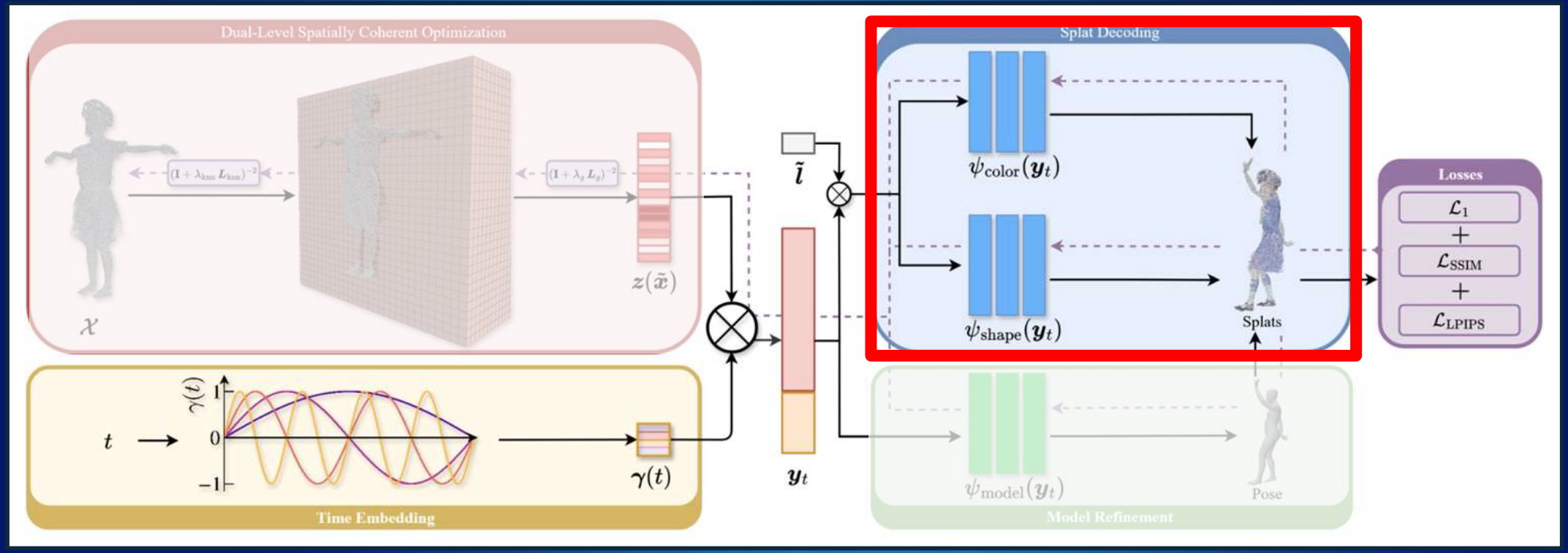
BONSAI
Keep the representation — optimize deployment

Reference Representation



PiG-Avatar: Hierarchical Neural-Field-Guided Gaussian Avatars (Kaltheuner et al. 2026)

What do we need for Deployment



PiG-Avatar: Hierarchical Neural-Field-Guided Gaussian Avatars (Kaltheuner et al. 2026)

Runtime Bottleneck

- Goal: preserve PiG-Avatar's canonical avatar representation and LoD hierarchy
- That means real-time decoding of up to 100k anchors, every animated frame.

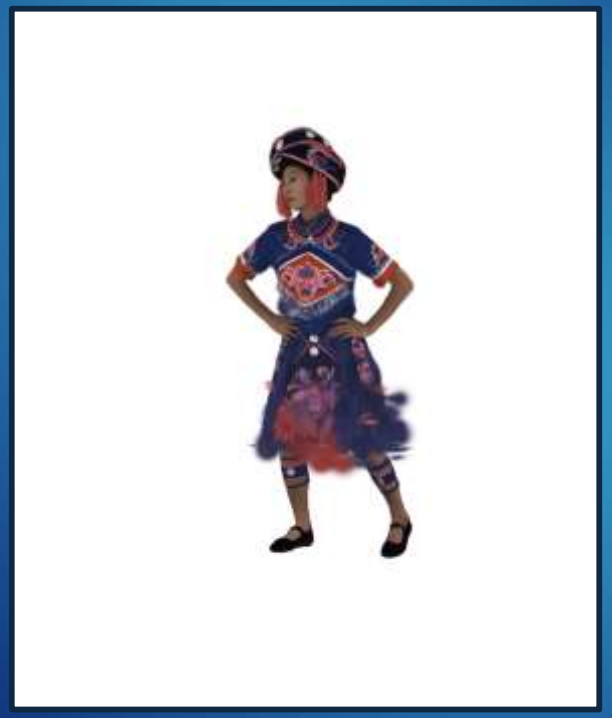
  → Baseline **WebGPU** implementation of full PiG-Avatar ran at **5 FPS** on a **RTX 4090** Desktop machine in our tests.
- How can we work around the Browser's constraints?



Why not retrain PiG-Avatar with smaller Decoders?



Reference(512)

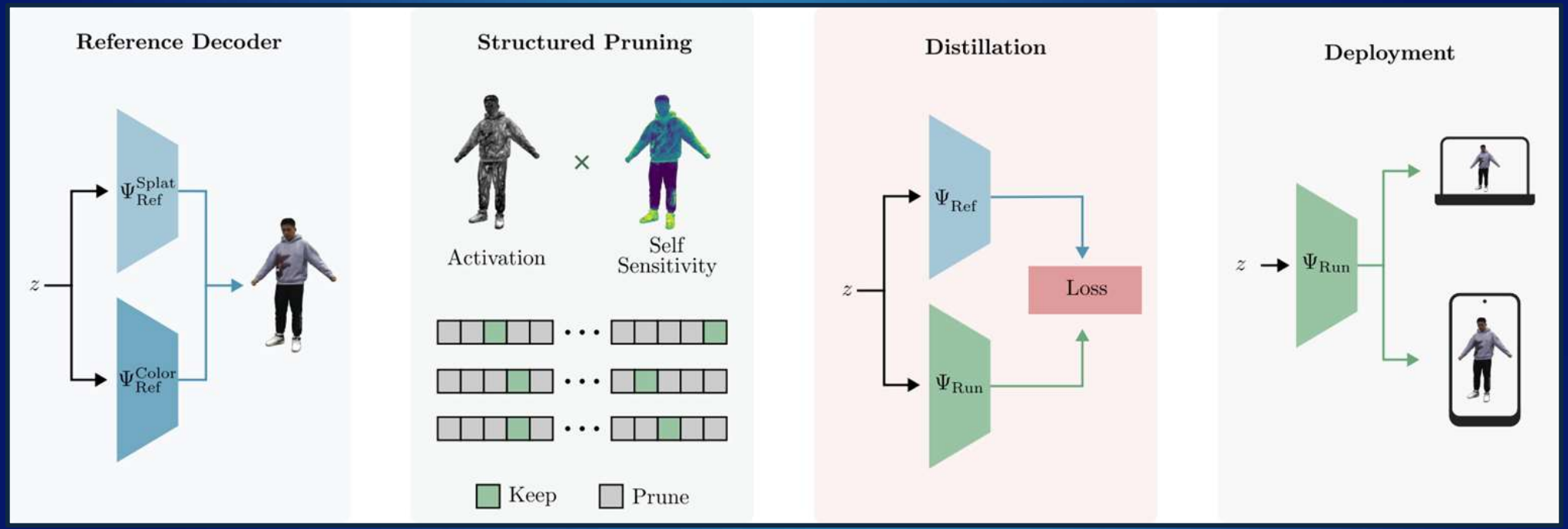


Retrained(64)

1. Quality does not degrade gracefully

2. Every target requires end-to-end retraining → dependent on data availability

BONSAI: Method Overview



512 → 64

hidden width

>98%

decoder parameter reduction

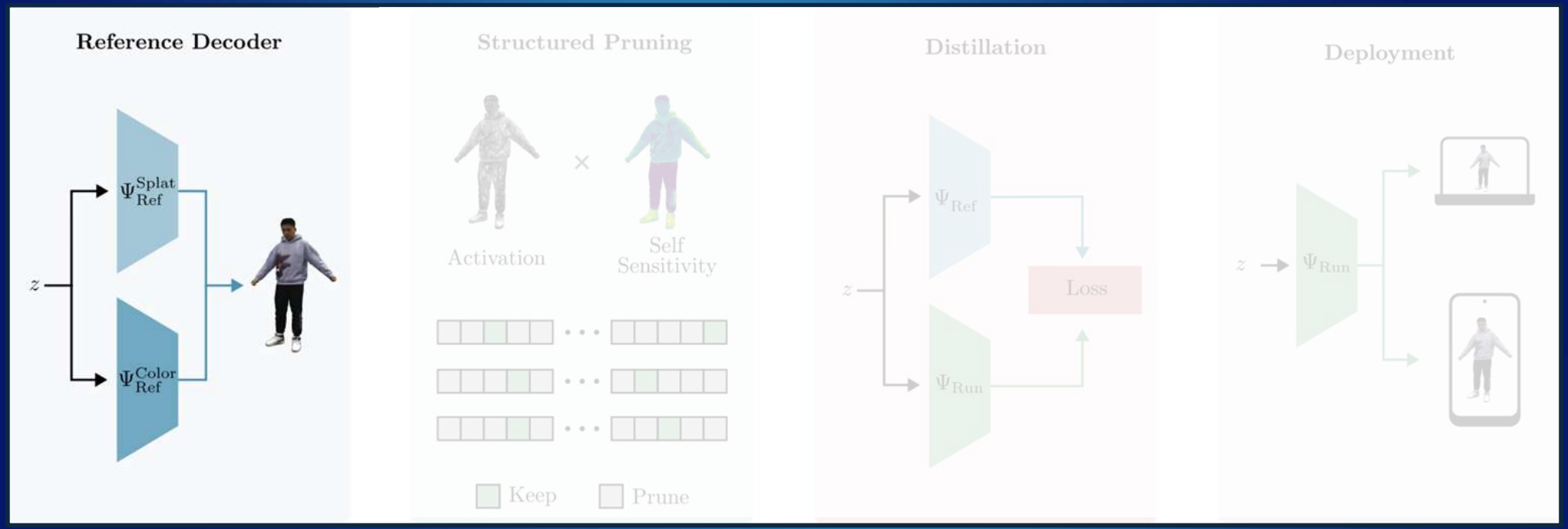
same

anchors + LoD hierarchy

client-side

WebGPU inference

BONSAI: Reference Decoder



2 x 512-wide MLPs **5 hidden layers each**

BONSAI: Reference Decoder

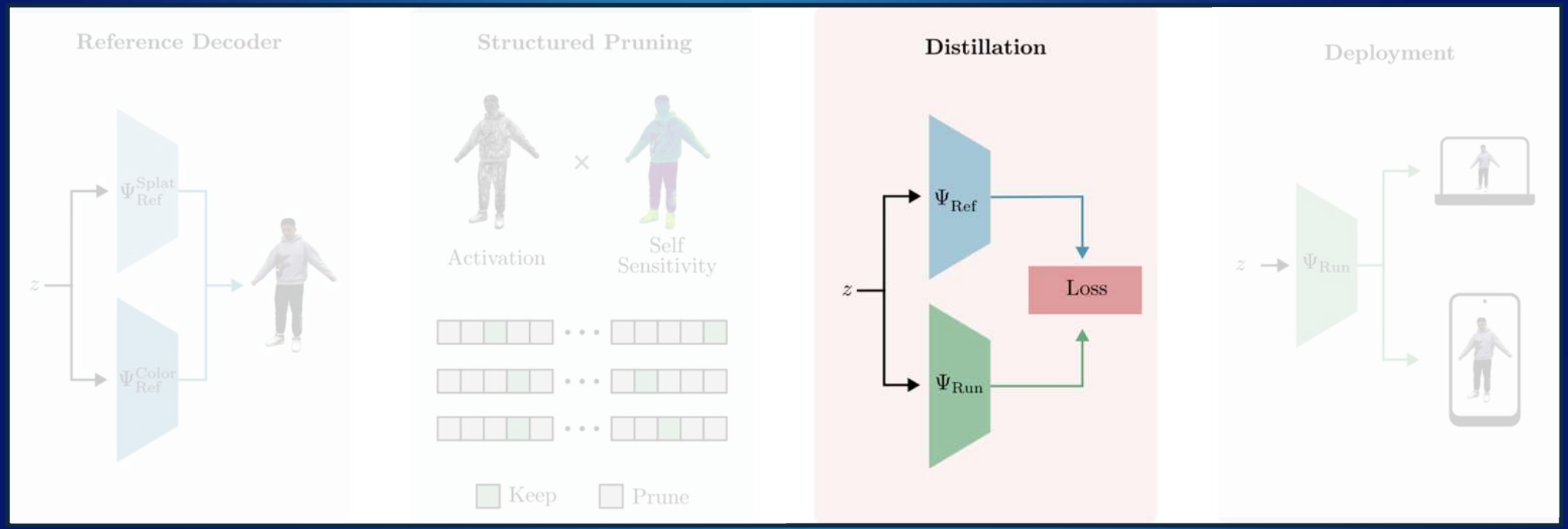
- We assume a trained neural Gaussian avatar represented by a set of canonical anchors $\mathbf{x}_i$ and latent codes $\mathbf{f}_i$
- Each anchor is attached to a parametric human model such as SMPL-X which provides a local frame and a kinematic transform for a pose $\boldsymbol{\theta t}$ at time $\mathbf{t}$
- For an input condition:

$$\mathbf{z}_{i,t} = [\mathbf{f}_i, \gamma(t), \phi(\theta_t), \ell]$$

where $\gamma\ (t)$ encodes time, $\phi\ (\theta t)$ encodes pose, and ℓ denotes the LoD level, a neural decoder predicts Gaussian attributes:

$$\Psi_{\text{Ref}}(\mathbf{z}_{i,t}) = \left(\Delta\boldsymbol{\mu}_{i,t}, \Delta\mathbf{q}_{i,t}, \mathbf{s}_{i,t}, o_{i,t}, \mathbf{c}_{i,t}\right)$$

BONSAI: Distillation



Reference Attribute Cache

- The Reference model is expensive, so instead of evaluating it repeatedly we:

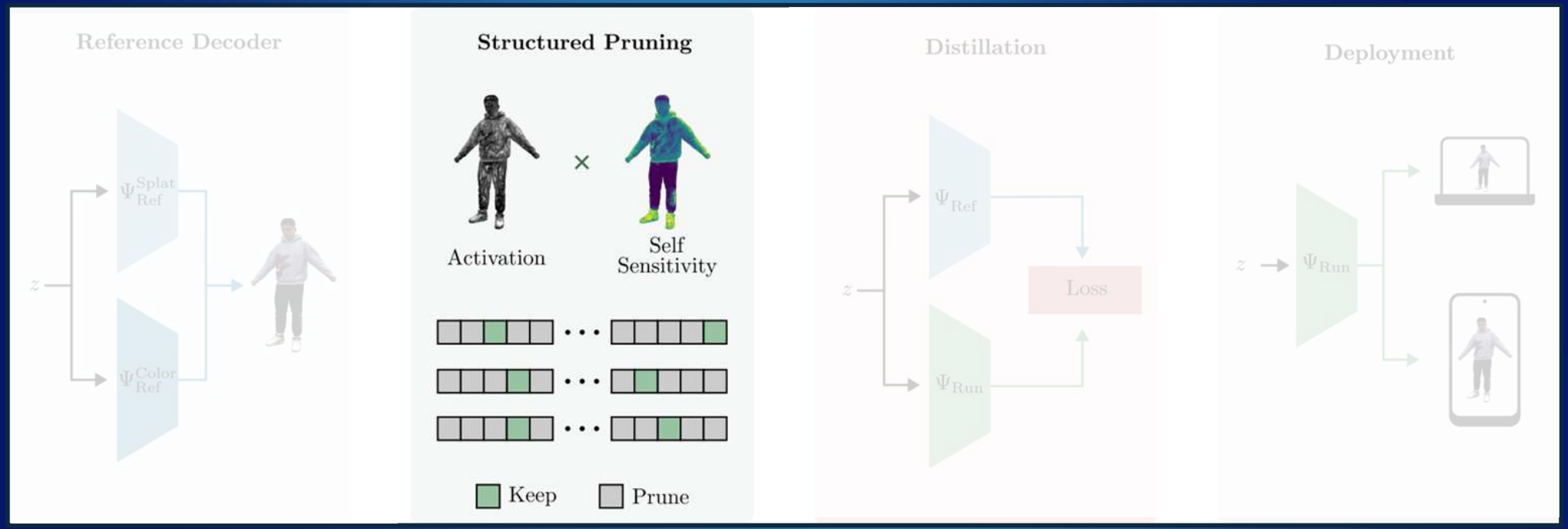
sample input conditions:
anchor • pose • time • LoD

evaluate reference once

$$\mathbf{y}_{i,t,\ell}^{\mathrm{Ref}} = \Psi_{\mathrm{Ref}}(\mathbf{z}_{i,t,\ell})$$

cache $\Delta\mu$, Δq, s, o, c

BONSAI: Structured Pruning



Reference-aware pruning

- How do we find optimal runtime decoder Initialization?
- Estimate the importance of each neuron per hidden layer and rank them accordingly then copy the top k = 64 neurons weights + biases

$$I_j^{(\ell)} = \left(\frac{1}{N} \sum_{n=1}^{N} \left| h_j^{(\ell)}(z_n) \right| \right) \left\| W_{\ell+1,j}^{\mathrm{Ref}} \right\|_2$$





Reference-aware pruning

- How do we find optimal runtime decoder Initialization?
- Estimate the importance of each neuron per hidden layer and rank them accordingly then copy the top k = 64 neurons weights + biases

$$I_j^{(\ell)} = \left(\frac{1}{N} \sum_{n=1}^{N} \left| h_j^{(\ell)}(z_n) \right| \right) \left\| W_{\ell+1,j}^{\mathrm{Ref}} \right\|_2$$

Frequently active over cached states





Reference-aware pruning

- How do we find optimal runtime decoder Initialization?
- Estimate the importance of each neuron per hidden layer and rank them accordingly then copy the top k = 64 neurons weights + biases





$$I_j^{(\ell)} = \left(\frac{1}{N} \sum_{n=1}^{N} \left| h_j^{(\ell)}(z_n) \right| \right) \left\| W_{\ell+1,j}^{\mathrm{Ref}} \right\|_2$$

Frequently active over cached states

×

Strong downstream influence

Self-Sensitivity Guided Neuron Selection

- Not every anchor affects the final rendering equally.

Perturb Gaussian attributes → Compare perturbed with reference render → Image space change





average across sampled poses and viewpoints

→ per-anchor self-sensitivity s_i

Self-Sensitivity Guided Neuron Selection

- Incorporating sensitivity weight $s_{i(\square)}$ into our importance score yields:

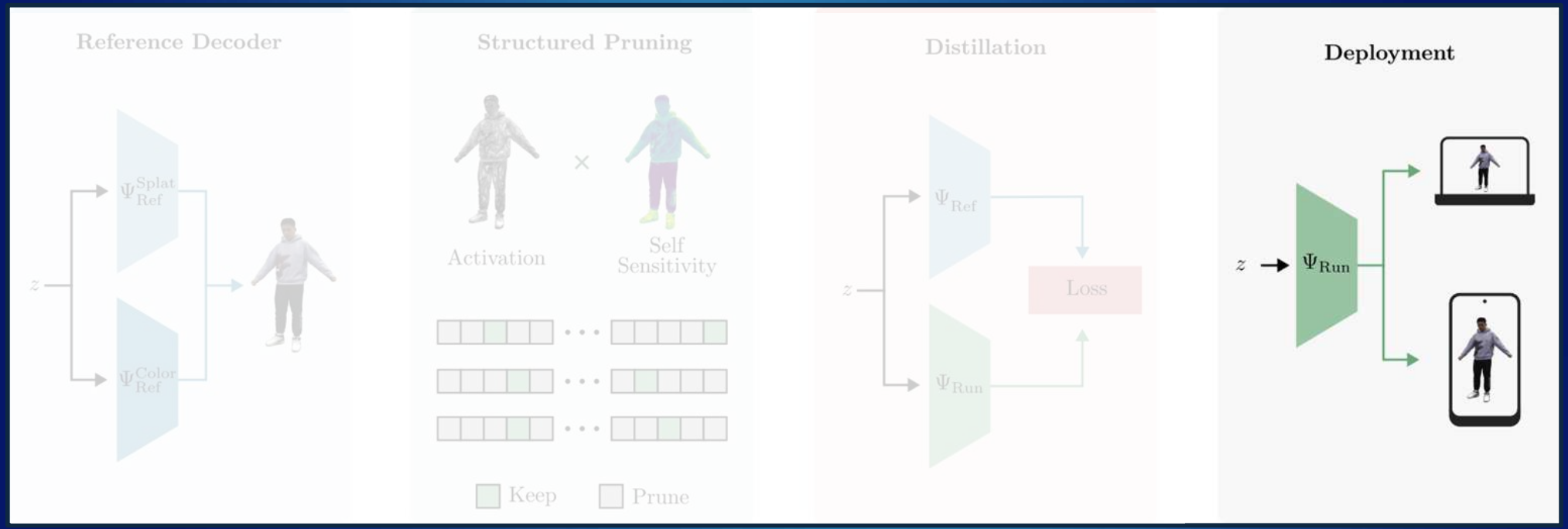
$$I_j^{(\ell)} = \left(\frac{1}{M} \sum_{n=1}^{M} s_{i(n)} \left| h_j^{(\ell)}(\mathbf{z}_n) \right| \right) \left\| W_{\ell+1,:,j}^{\mathrm{Ref}} \right\|_2$$





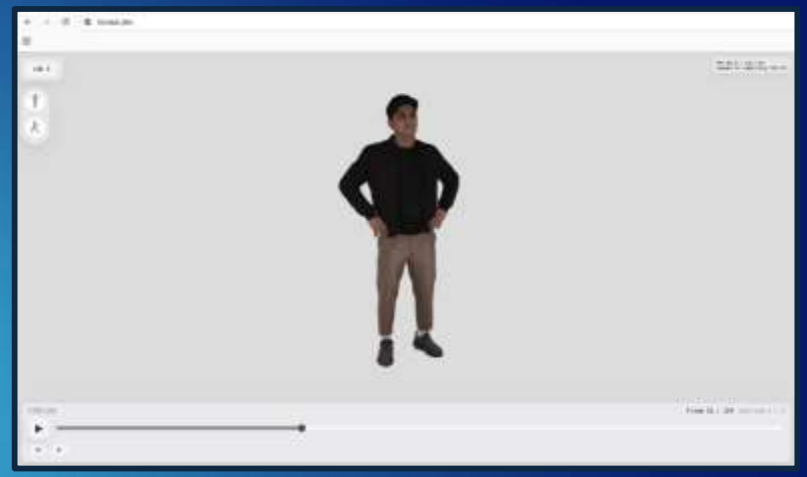
- Visual sensitivity biases the model capacity toward anchors where errors are most noticeable.

BONSAI: Deployment



WebGPU Runtime



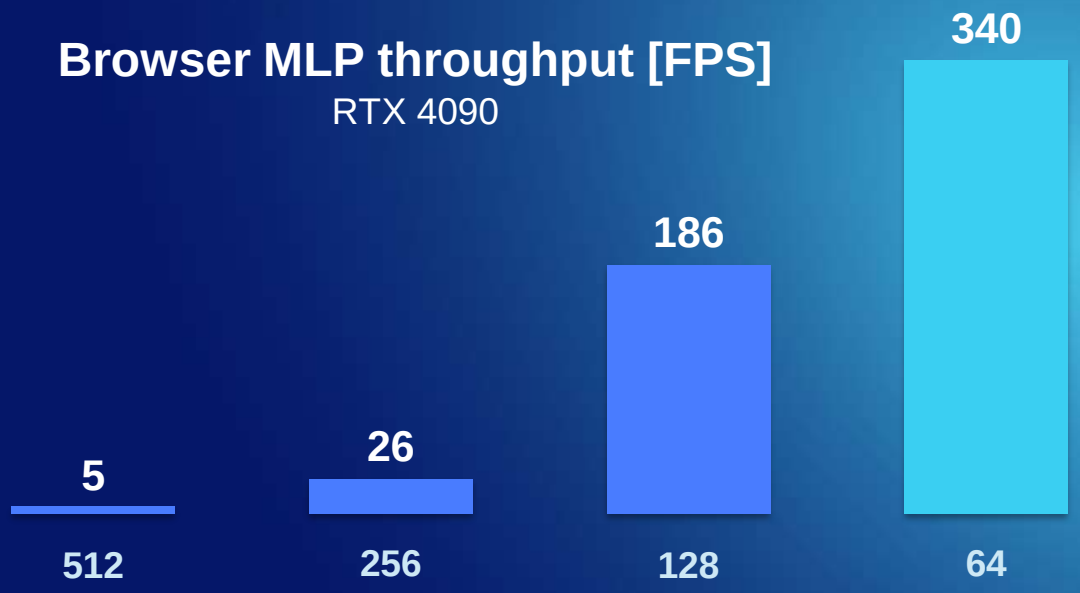


- Layer-by-layer MLP inference repeatedly transfers intermediate activations through GPU memory.

  → **Fused inference** keeps intermediate activations on-chip, substantially reducing memory traffic.

- **Shared-memory weight caching**: We load layer weights once per workgroup to reduce global memory traffic; feasible for widths ≤64 within the 16 kB WebGPU limit.

Main Quantitative Results



512 → 64 width
5 FPS → 340 FPS

Model size
(anchors + MLP weights)
46 MB → 39 MB

Why?
Storage is dominated by Gaussian data, not decoder weights.

Compression changes runtime cost far more than asset size

Runtime fusion further reduces per-frame cost

UNIVERSITÄT BONN

Device	optimization	SMPL-X Attachment	Splat MLP	Splat Preprocessing	Radix Sort	Draw	Total GPU Time [ms]
RTX 1050	unfused	1.99	52.54	0.27	0.87	1.25	56.93
RTX 3080	unfused	0.59	7.91	0.46	0.29	0.42	9.67
RTX 4090	unfused	0.45	2.53	1.30	0.47	0.34	5.02

- Even after compression, MLP inference remains the dominant frame-time component.



Runtime fusion further reduces per-frame cost

UNIVERSITÄT BONN

Device	optimization	SMPL-X Attachment	Splat MLP	Splat Preprocessing	Radix Sort	Draw	Total GPU Time [ms]
RTX 1050	fused	2.02	51.27	0.26	0.88	1.22	55.65
RTX 3080	fused	0.30	4.45	0.20	0.13	0.69	5.77
RTX 4090	fused	0.44	2.29	0.20	0.11	0.38	3.42

- Even after compression, MLP inference remains the dominant frame-time component.



Runtime fusion further reduces per-frame cost

UNIVERSITÄT BONN

Device	optimization	SMPL-X Attachment	Splat MLP	Splat Preprocessing	Radix Sort	Draw	Total GPU Time [ms]
RTX 1050	fused shared memory	2.07	28.30	0.26	0.89	1.22	32.75
RTX 3080	fused shared memory	0.31	2.60	0.23	0.14	0.71	3.99
RTX 4090	fused shared memory	0.50	1.62	0.25	0.13	0.44	2.94

- Even after compression, MLP inference remains the dominant frame-time component.



Visual Fidelity of the Runtime Decoder

UNIVERSITÄT BONN

	PSNR↑	SSIM↑	LPIPS↓
THUman4.0	35.06 dB	0.9894	0.0117
DNA-Rendering	32.33 dB	0.9918	0.0173

Evaluation is against the corresponding 512-wide reference decoder, isolating conversion error.

Visual Fidelity of the Runtime Decoder

UNIVERSITÄT BONN

Naive Retrained (64)





BONSAI (64)

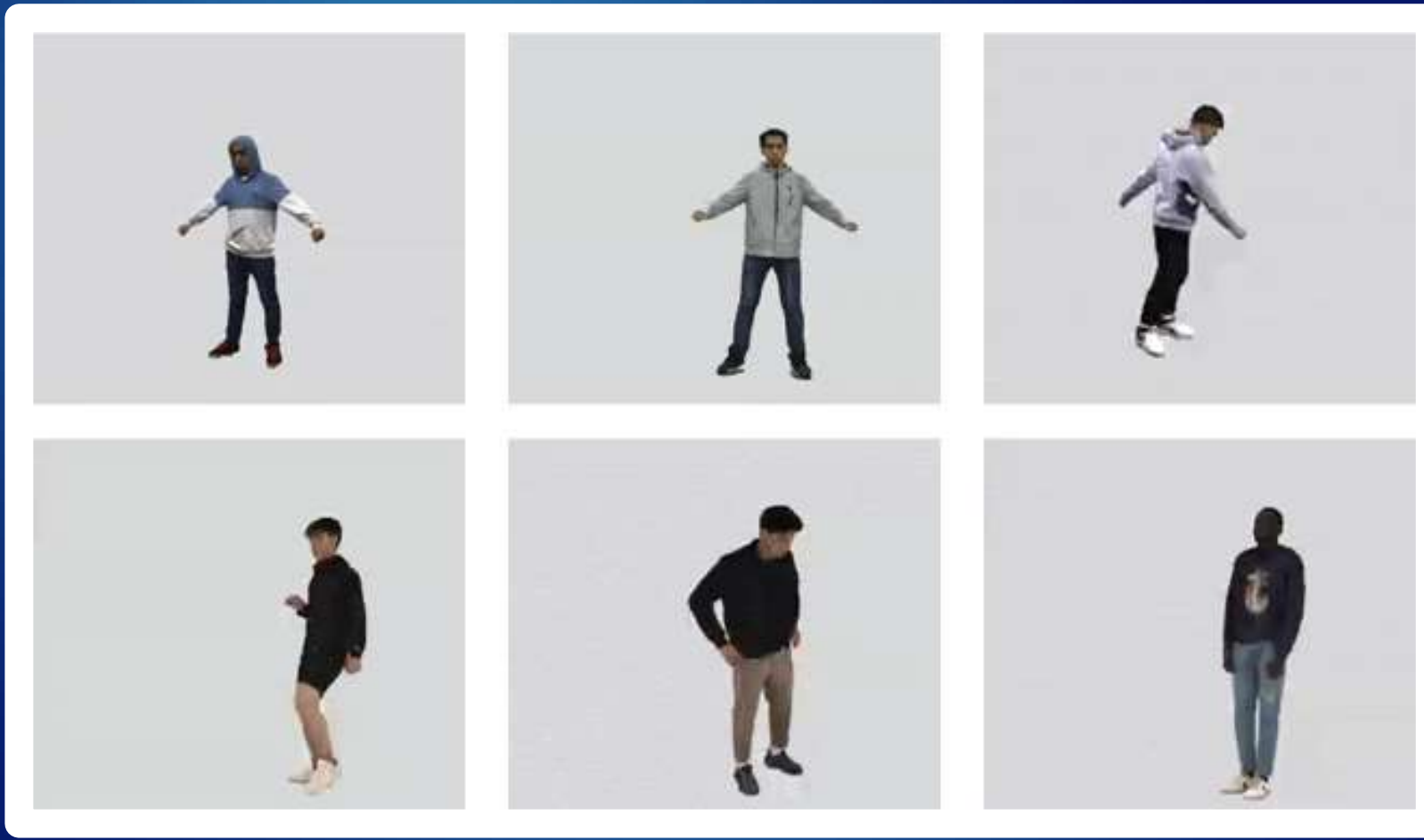




Main Qualitative Results

THUman4.
0

DNA-Rendering



Progressive LoD Streaming

LoD	Anchors	Data MB	TTFF ms	Δ LOD0 ms
0	12.5k	4.90	2188.0	-
1	25k	+ 4.90	2193.0	5.0
2	50k	+ 9.80	2229.3	41.3
3	100k	+ 19.60	2241.6	53.6

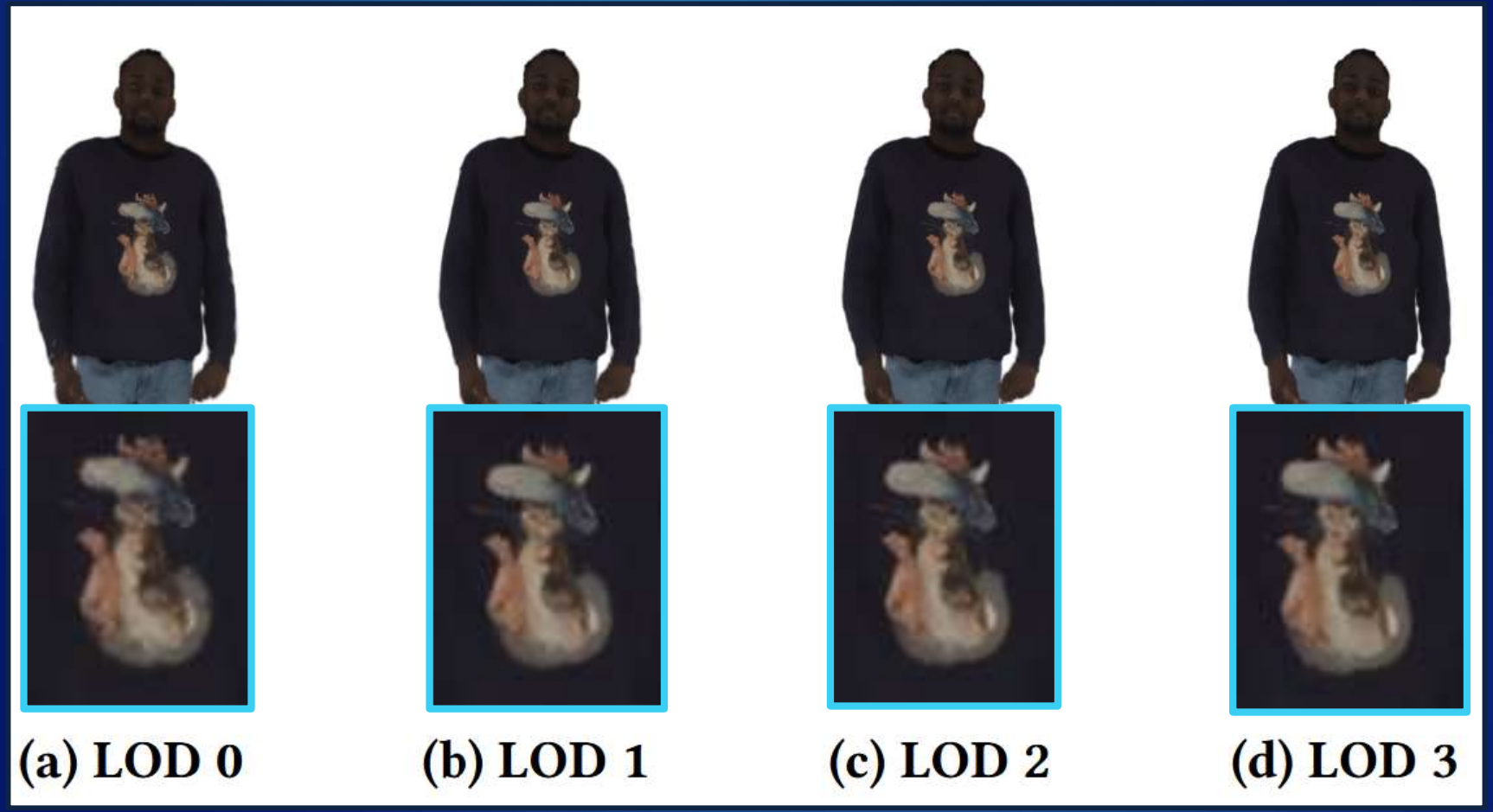
- Decoder compression reduces per-anchor inference cost.
- Progressive LoD additionally reduces the number of active anchors, lowering inference and rendering cost while enabling progressive delivery.

Progressive LoD Enables Mobile Playback

Influence of LoD on mobile total GPU time.

Device	LoD 0 [ms]	LoD 1 [ms]	LoD 2 [ms]	LoD 3 [ms]
iPhone 15	31.85	61.90	123.06	274.54
Honor Magic 6 Pro	39.14	69.05	129.31	231.10
Samsung A73	28.25	41.73	68.03	122.35

Qualitative LoD Comparison



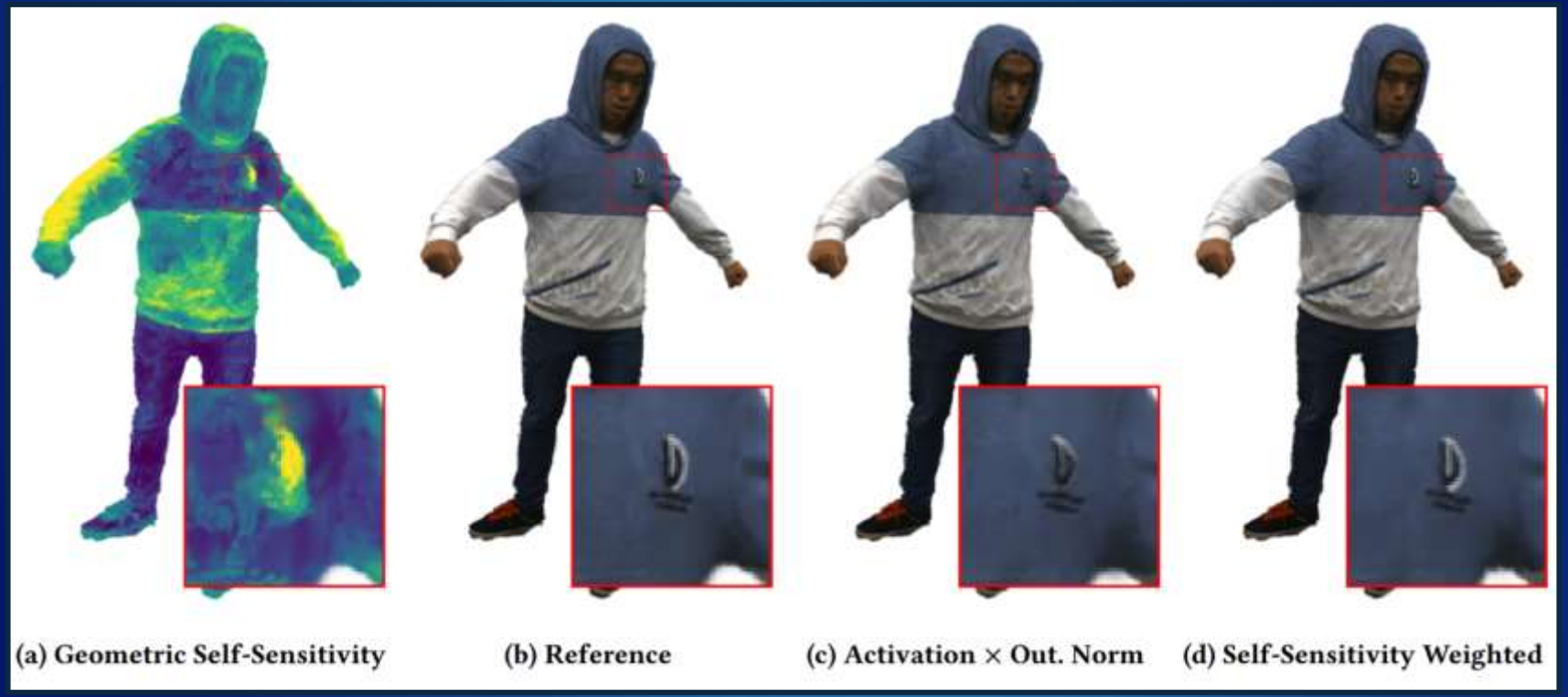
Which pruning strategy best preserves quality?

Evaluated on THUman4.0 for LOD 3 (N=100k)

Importance score	PSNR↑	SSIM↓	LPIPS↑
Random	34.14	0.9878	0.0128
Activation × out. norm	34.85	0.9890	0.0119
Self-sensitivity (ours)	35.06	0.9894	0.0117

Same 64-wide architecture, objective, and training budget, only initialization changes.

Which pruning strategy best preserves quality?



Same 64-wide architecture, LoD 3, objective and training budget, only initialization changes.

Discussion

- BONSAI is **not** a reconstruction method. We **cannot** recover detail absent from the reference avatar.

- Other Avatar representations remain future work.

Core idea: separate avatar authoring from deployment.
Build the best avatar first, then derive the runtime representation you can afford.

Conclusion

512 → 64
decoder width

5 → 340 FPS
browser MLP throughput

~2.2 s
time to first coarse avatar

BONSAI makes an existing neural Gaussian avatar Web-ready instead of inventing a new avatar representation.

Structured pruning + distillation solve neural inference; fused WebGPU + progressive LoD solve execution and delivery.

IN.NRW INNOVATIONSFÖRDERAGENTUR
Ministerium für Wirtschaft, Industrie, Klimaschutz und Energie des Landes Nordrhein-Westfalen
Kofinanziert von der Europäischen Union
UNIVERSITÄT BONN



Thank You!

Web3D 2026

Doha, Qatar

BONSAI
Browser-Optimized Neural Gaussian Avatar Streaming

kitz@cs.uni-bonn.de
linkedin.com/in/sina-kitz

