SIGGRAPH 2026
Los Angeles 19–23 JUL

The Premier Conference & Exhibition on
Computer Graphics & Interactive Techniques

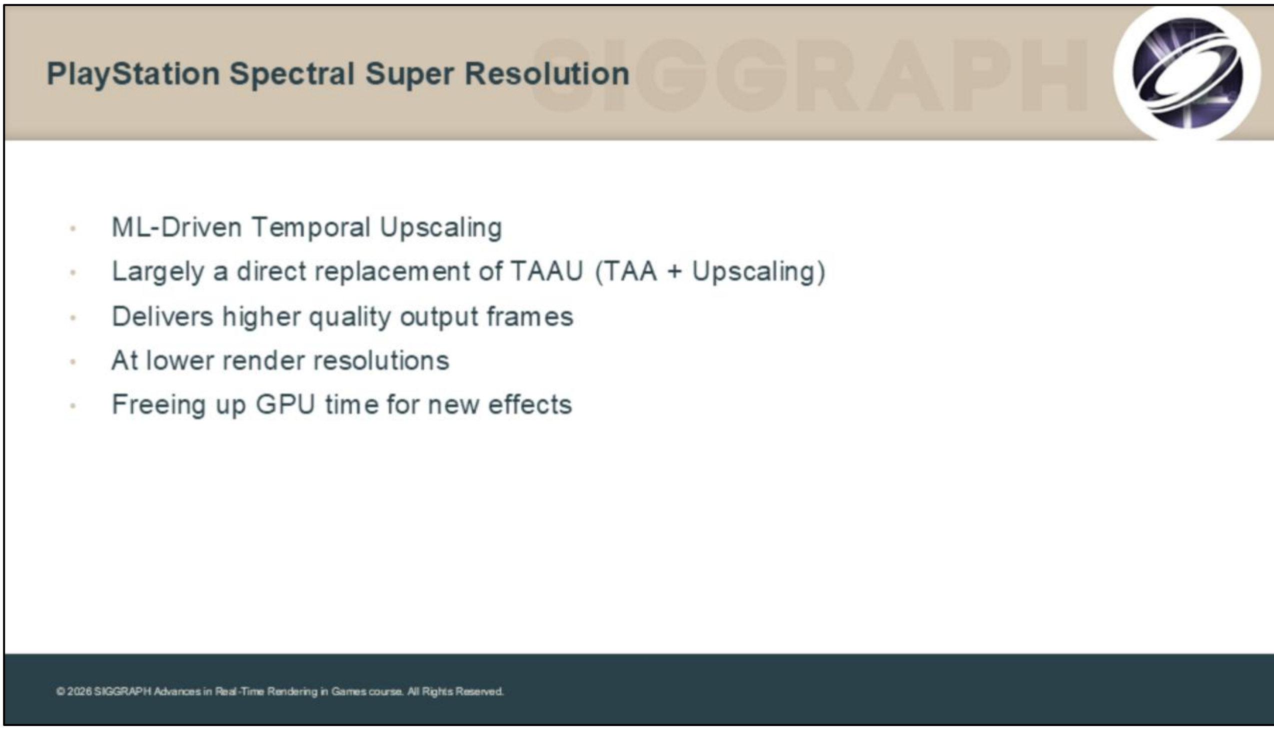
Upgrading PSSR on PS5 Pro

2026

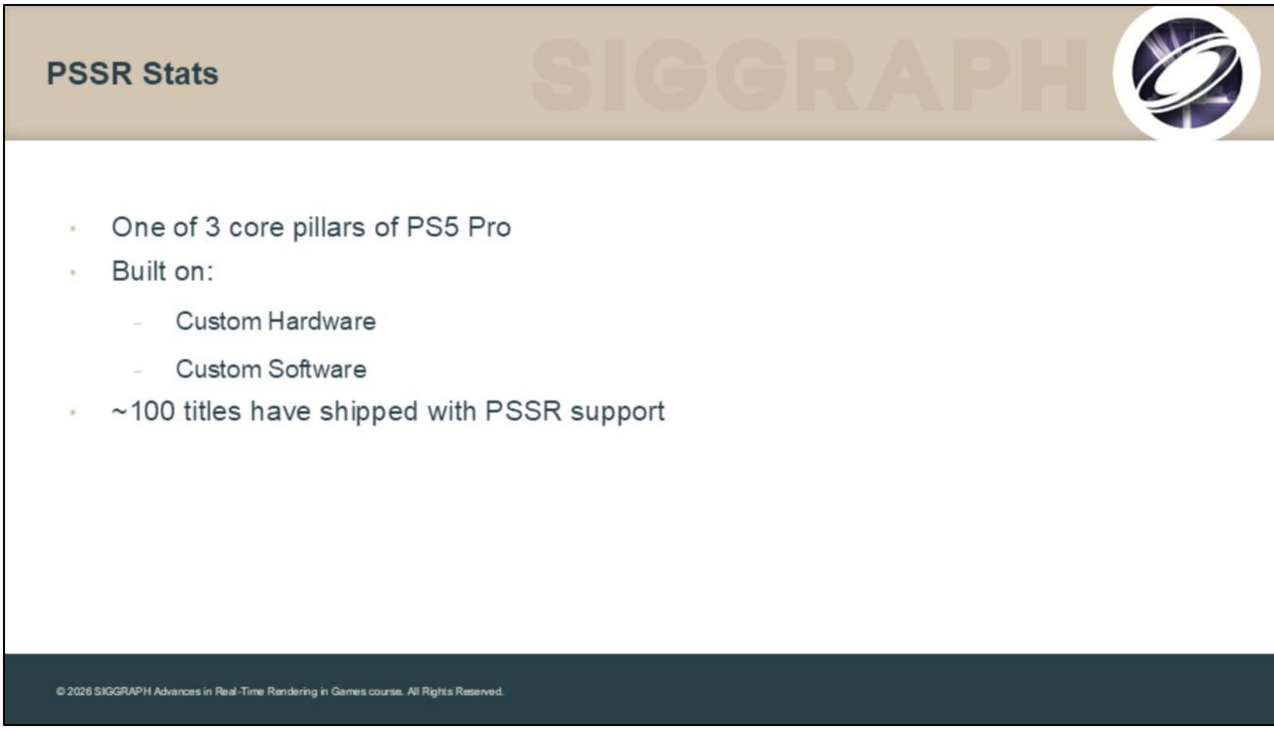
- This talk covers the upgrade to PlayStation Spectral Super Resolution (PSSR), released earlier this year.

The Plan

- Background / Introduction
  - Why all of this?
- Launch PSSR: Our first step
  - A bit of post-mortem
- Upgraded PSSR: Overview of Key Changes
  - What changed and why it worked



- The goal is not only higher quality than TAAU, but comparable quality at a lower render resolution, letting titles reclaim the GPU time saved.



- Referenced in Mark Cerny's 2024 PS5 Pro technical deep dive. Built on custom hardware, software, trainer, and dataset. The ~100 titles include both the launch and upgraded versions.

Upgraded PSSR

- A research collaboration with AMD (Project Amethyst)
  - The implementation is custom for PlayStation hardware
  - But based on the shared research and training
- Replaces the launch PSSR version
- System-level override allows use in shipped titles
  - Override applies to all titles
  - Without a title update

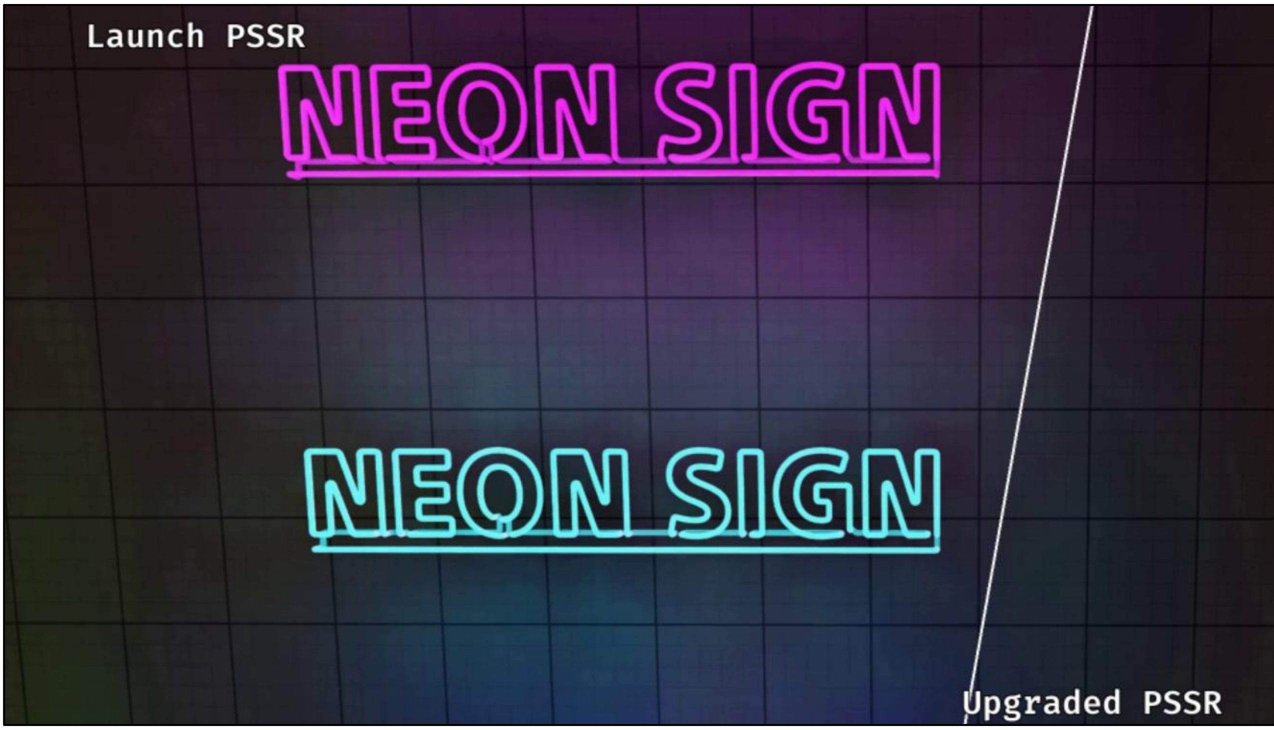
- Project Amethyst is a research collaboration with AMD; the upgraded model is a custom PlayStation implementation built on that shared research and training. Using the updated library directly gives the best quality and performance, while the system-level override upgrades already-shipped titles in place without a title update.

Improvements

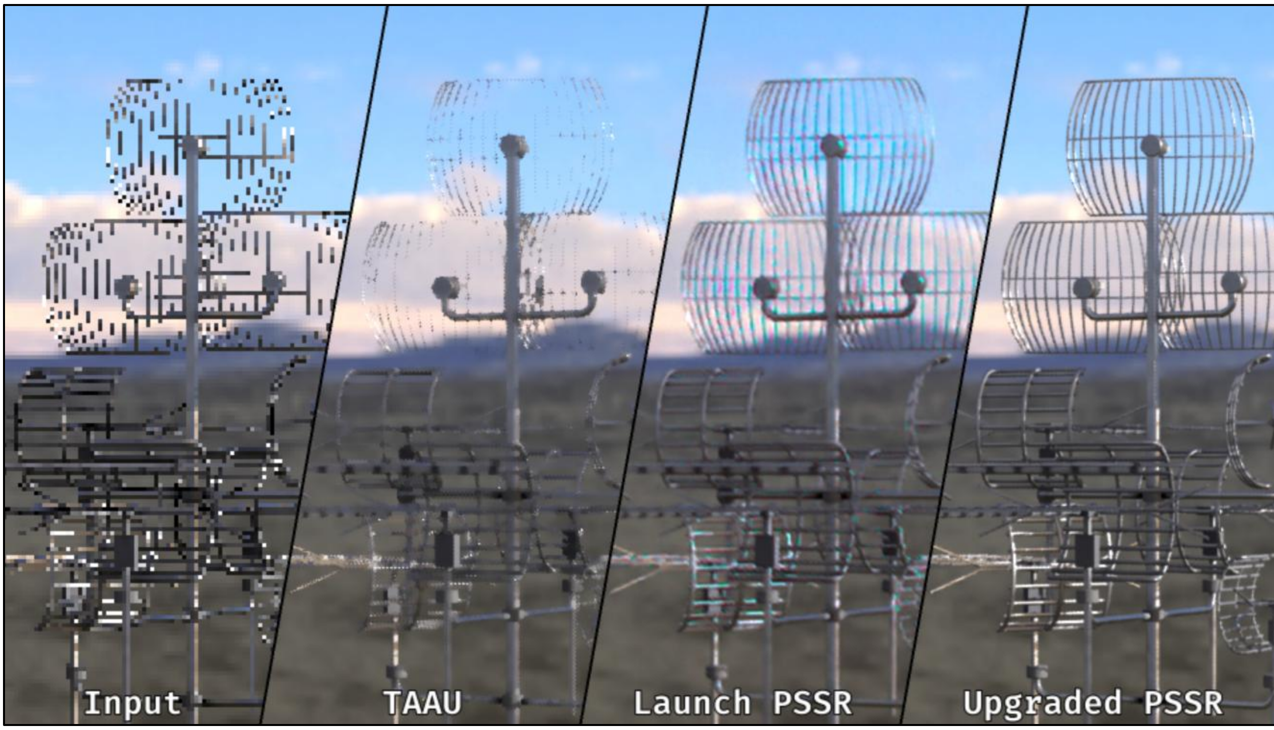
- Improved temporal stability
- Improved details / sharpness
- Less memory
- Less GPU time

Sizzle Reel!

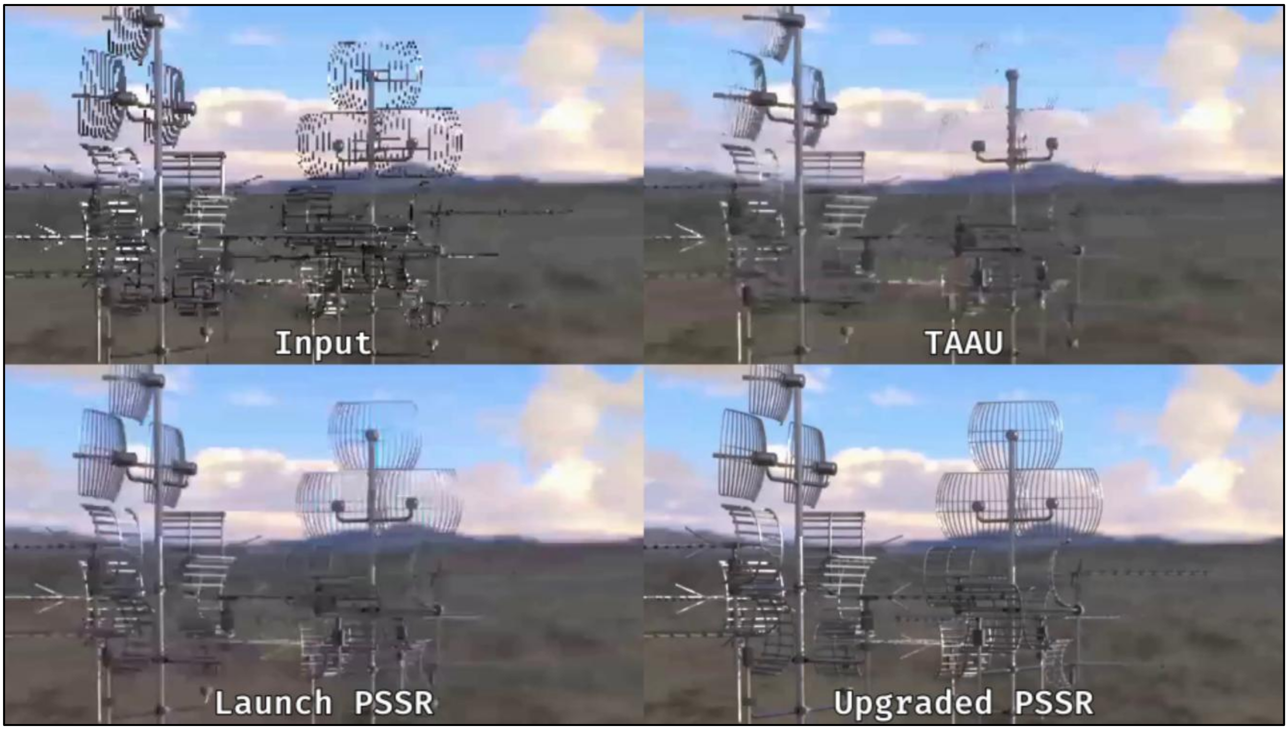
- No third-party footage is shown; all comparisons use first-party demo content. Independent outlets and the official Sony blog have published additional before/after comparisons.



- Headline improvement is temporal stability, especially for low-frequency lighting updates. Left is Launch PSSR; right is Upgraded.

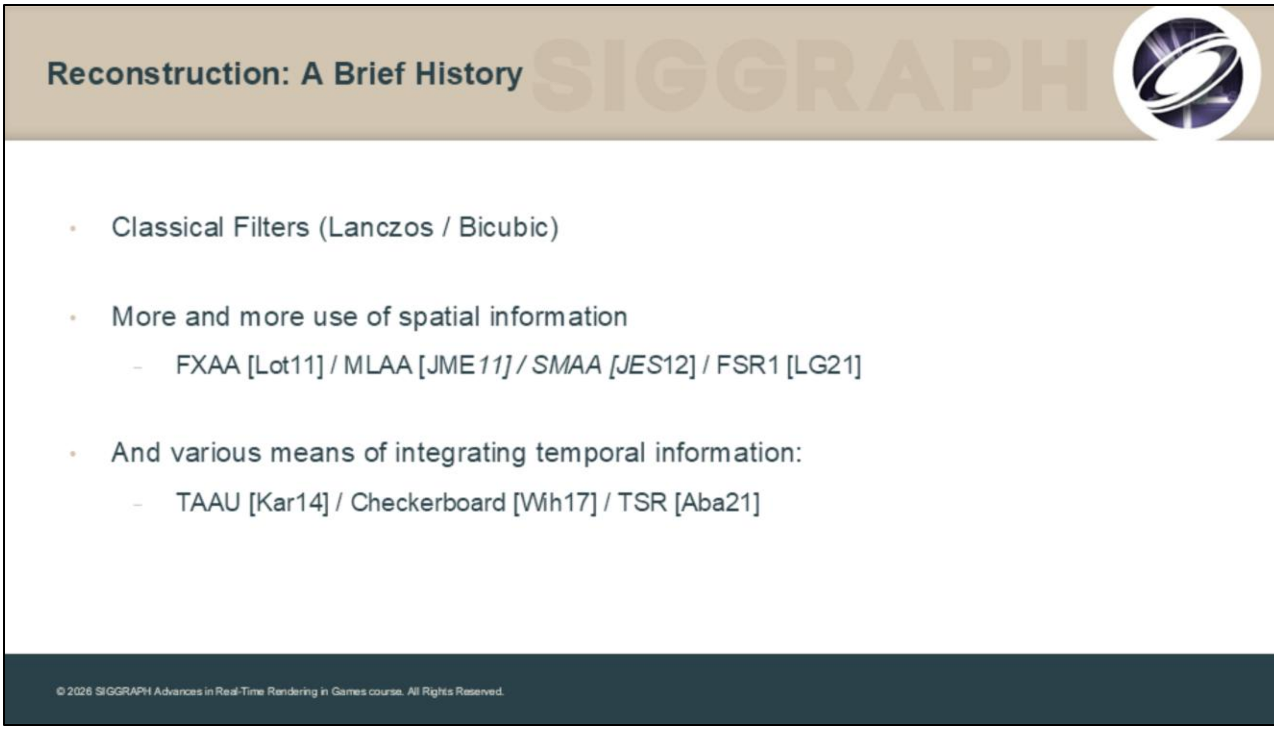


- A 3x upscale (720p input, 2160p output). The static comparison shows a clear increase in sharpness, among other improvements covered later.

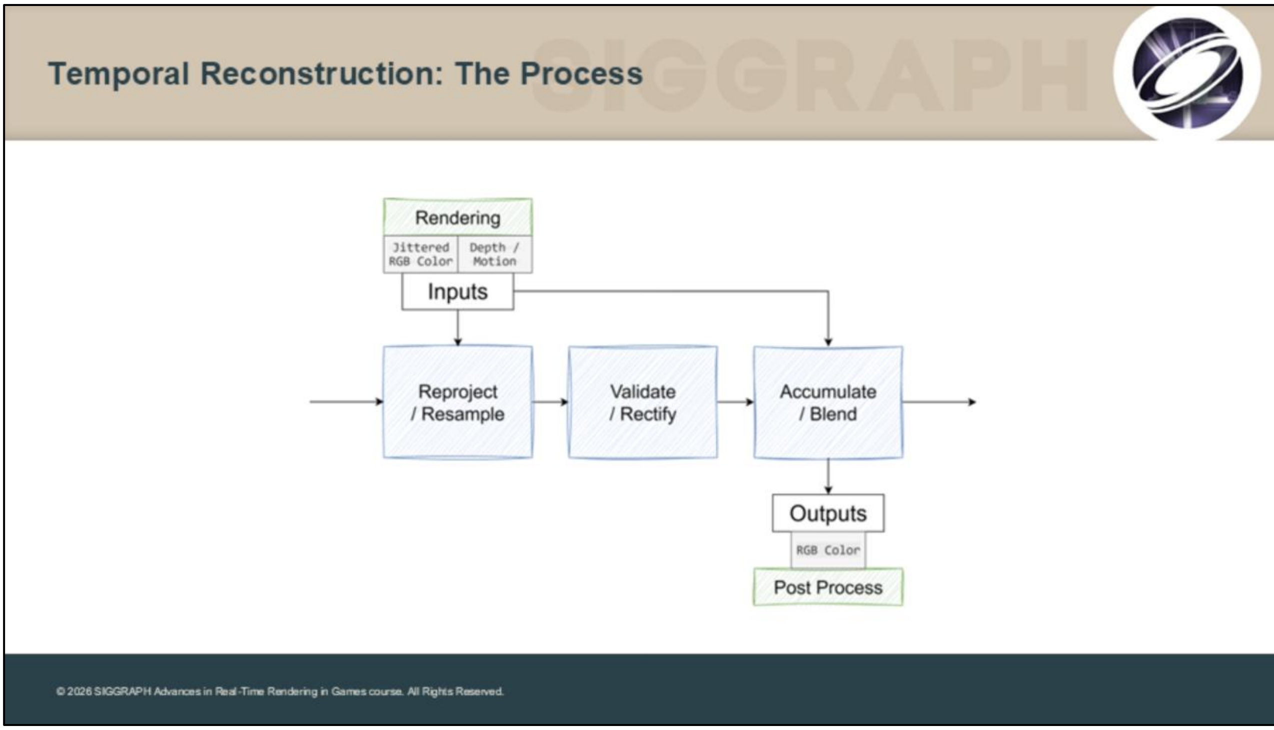


- Stability is also improved here.

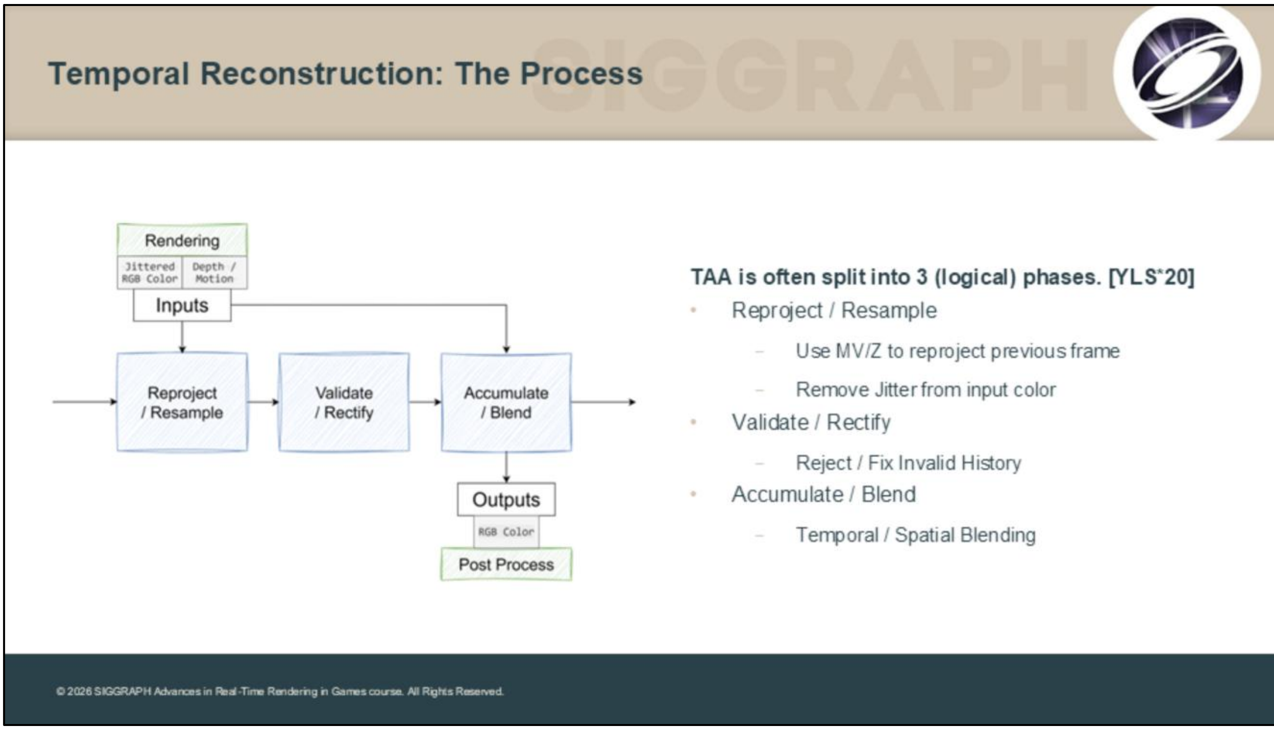
- Framing for the section: ML is presented as a natural continuation of a long line of reconstruction algorithms, which is the mental model behind the approach taken.



- Upscaling and temporal anti-aliasing are now near-universal in AAA titles, which makes them a high-value target and raises the question of whether ML meaningfully helps here.



- Temporal reconstruction reuses reprojected history frames to add usable information, at the risk of introducing artifacts.



- TAA needs only a few inputs (jittered color, depth, and motion vectors), though some implementations use more. The three logical phases are shown on the slide.

Heuristics Abound

Many steps are classical filters

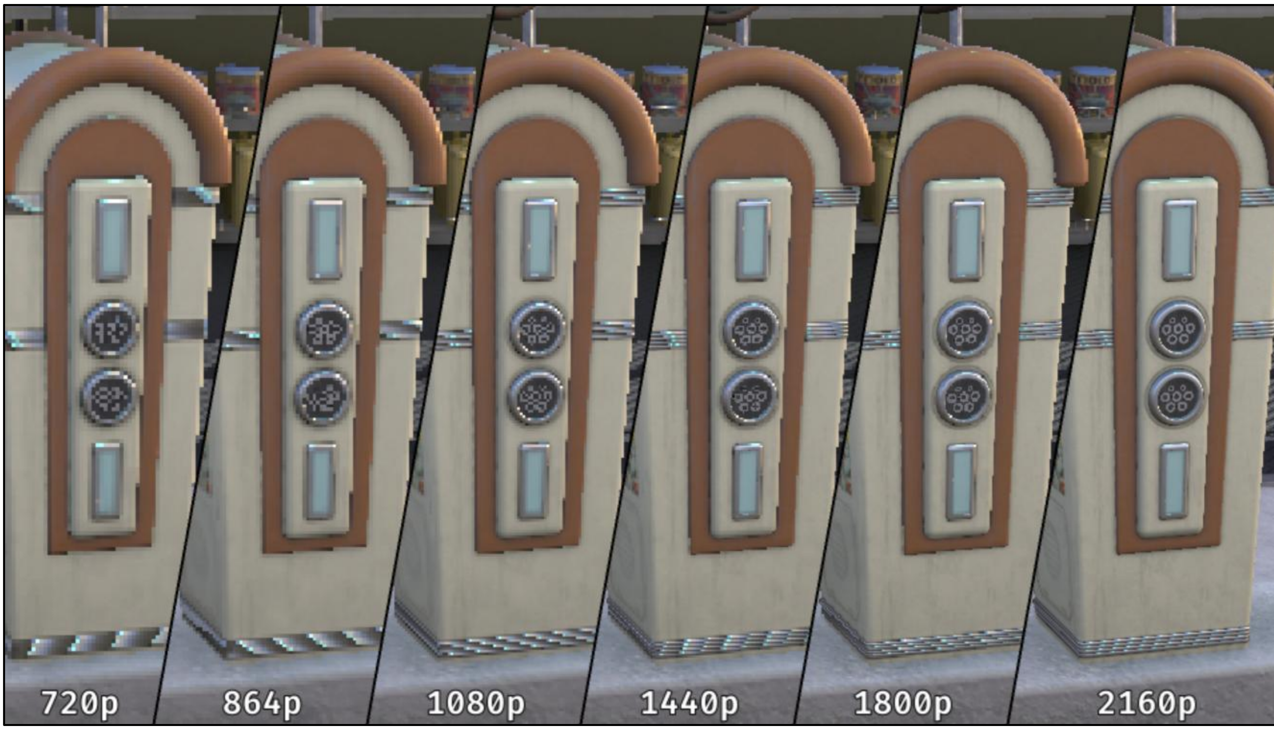
But temporal and spatial blending require decisions

- Reject / Rectify History?
- Antialias Edges?
- Remove Fireflies? Leave Fireflies?

These decisions are often based on:

- Hand tuned pattern and edge detection
- Clever / opportunistic optimizations

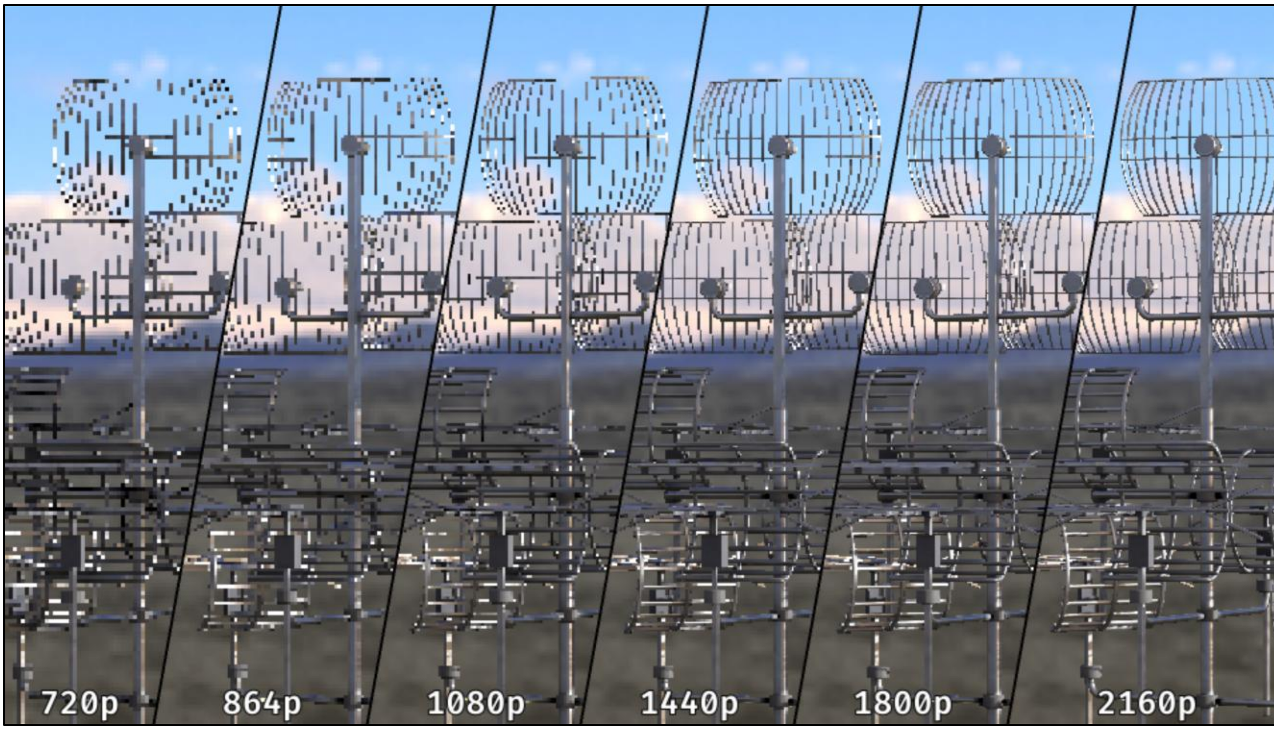
- Most TAA steps are resampling and filtering; only a few require decisions with conflicting goals. Yang et al.'s survey describes history rejection and rectification as "a challenging and ill-posed task." Hand-tuned heuristics have shipped many games successfully, but both history and input color are becoming less reliable.



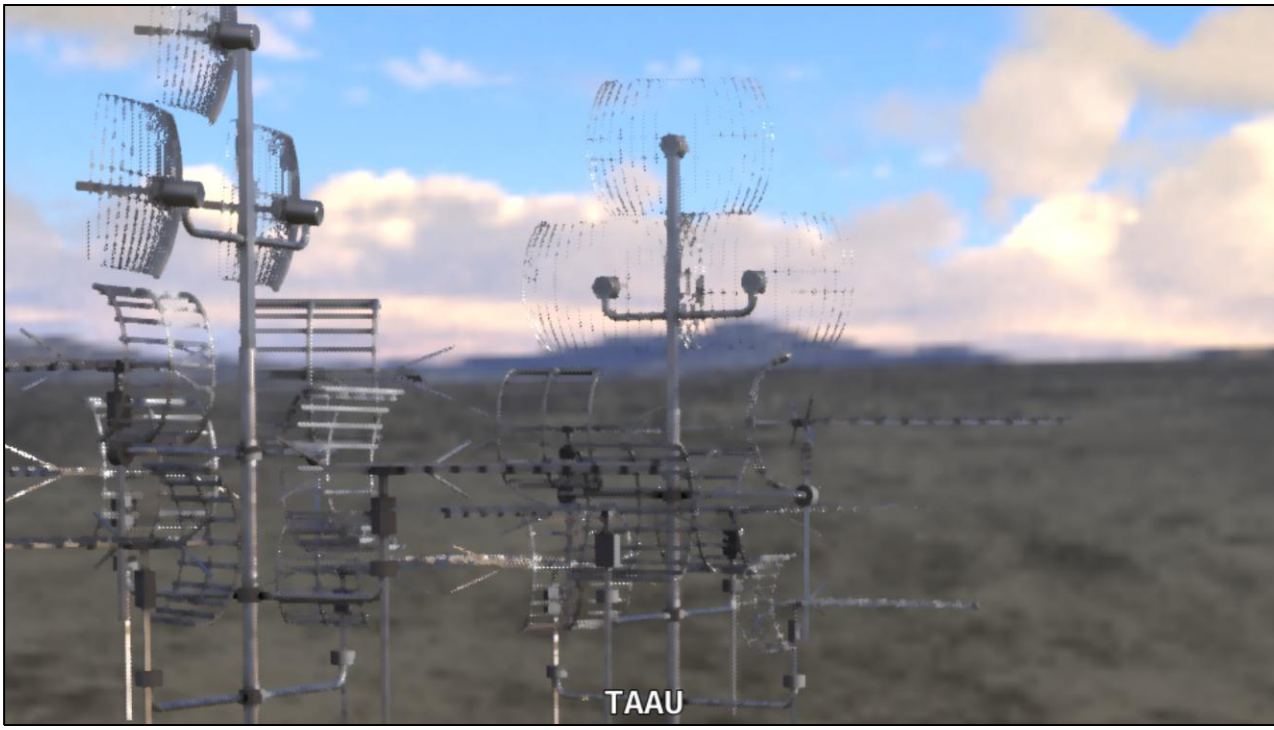
- Raw input color at decreasing input resolutions. As resolution drops, highly detailed content becomes hard to distinguish from noise in a single frame, and temporal patterns spread farther apart, making TAA's decisions harder. TAAU was not designed for 3x upscale, but that is the direction games are moving.

- Even in a still image, breakup appears in high-detail areas.

- In motion, the issues are temporal as well as spatial: history is hard to balance with so little, noisy information.



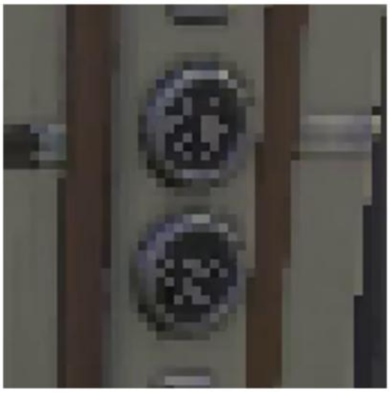
- Lower input resolution means less information per frame and less robust choices.



- Noisy input again causes issues.

- In motion the temporal breakup is clear; edge-reconstruction heuristics do not hold up at this upscale.

Content Is Becoming Noise



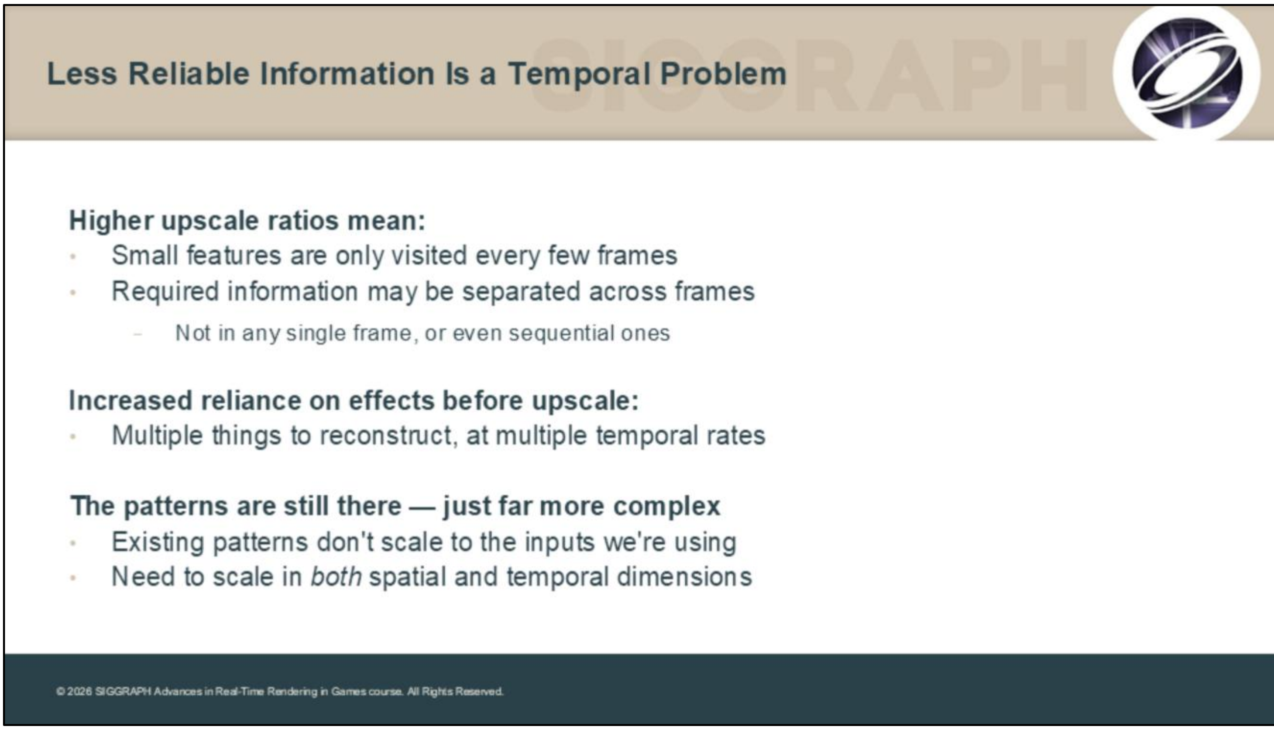
Content is quickly and intentionally approaching noise

- Lower resolution renders (with aggressive mip bias)

While also becoming less coherent and less reliable

- Particles / volumetrics / translucency
- Stochastic Effects
- Periodic temporal updates (lighting)
- RT Effects
- Inconsistent jitter

- Input is approaching noise for many reasons beyond low resolution. TAA is designed to resolve some of these (jitter, stochastic alpha); others were historically handled after TAA but are increasingly done before it to save GPU time.



- As upscale ratios rise, the temporal incidence rate (how often related information appears) drops. Required detail may be spread across non-sequential frames, and pre-upscale effects push it farther apart. The patterns still exist but are far more complex, and must scale in both space and time.

A Way Forward

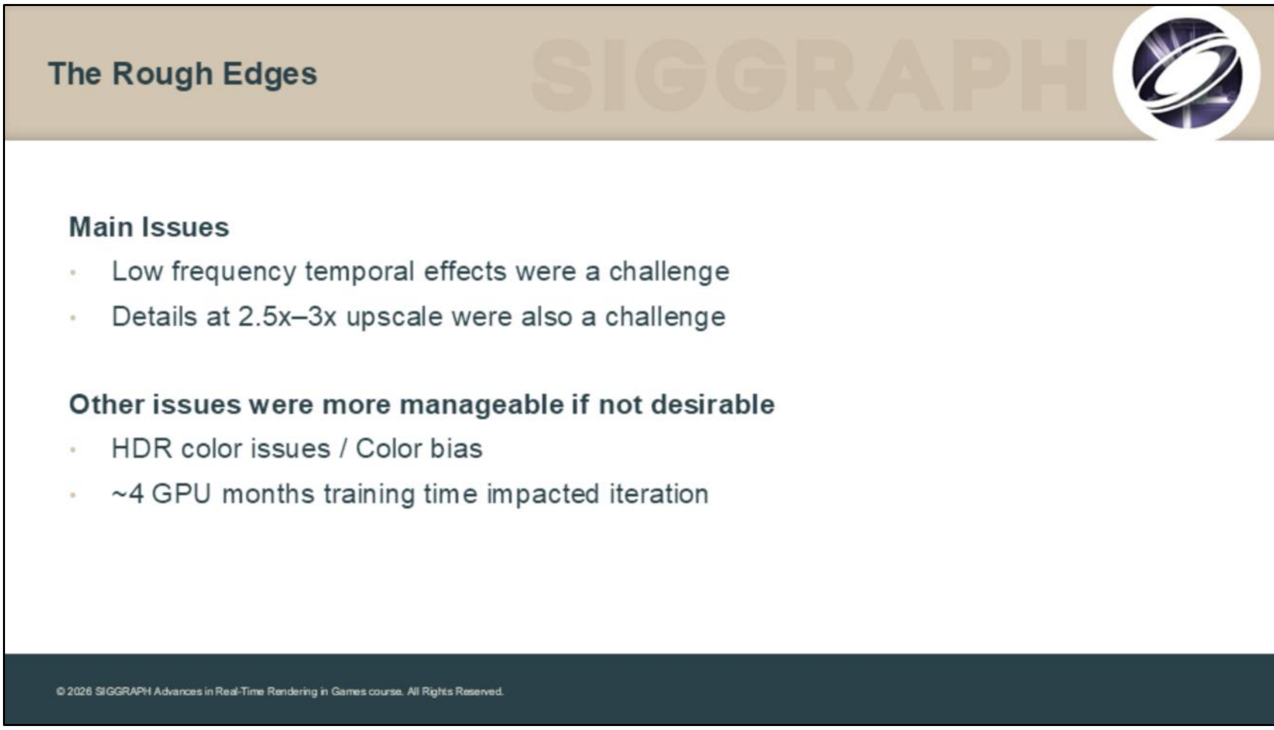
- Need orders of magnitude more complexity
  - Where TAA may be managing ~10 patterns
  - I feel like we need 100s or 1000s of patterns
  - Along with a complex state machine / decision tree
- Cost is also an issue — (It can't be 100x more expensive)
  - Need to stay highly HW-efficient

- Launch PSSR development began over five years ago, when even PC solutions were early. Rather than targeting existing hardware, the hardware architecture and instruction set were designed specifically for this model.

The Promise

- "Quality Mode" at "Performance Mode" / 60fps
- And it delivered in many titles
- Initial reviews were positive

- The promise: "quality mode" visuals (from PS5 base, 30fps) at "performance mode" speed (60fps) on PS5 Pro, a 2x performance gain with similar or improved visuals. It delivered across many titles, with positive initial reviews, though some rough edges remained.



- The main challenges were temporal stability (notably low-frequency effects like RTGI) and sharpness at 2.5x-3x upscale; 2.5x proved more common than expected despite not being an official PC ratio. Secondary items were HDR color representation and a ~4 GPU-month training time that slowed iteration.

- Slow, low-frequency temporal changes were difficult for Launch PSSR to address without side effects.

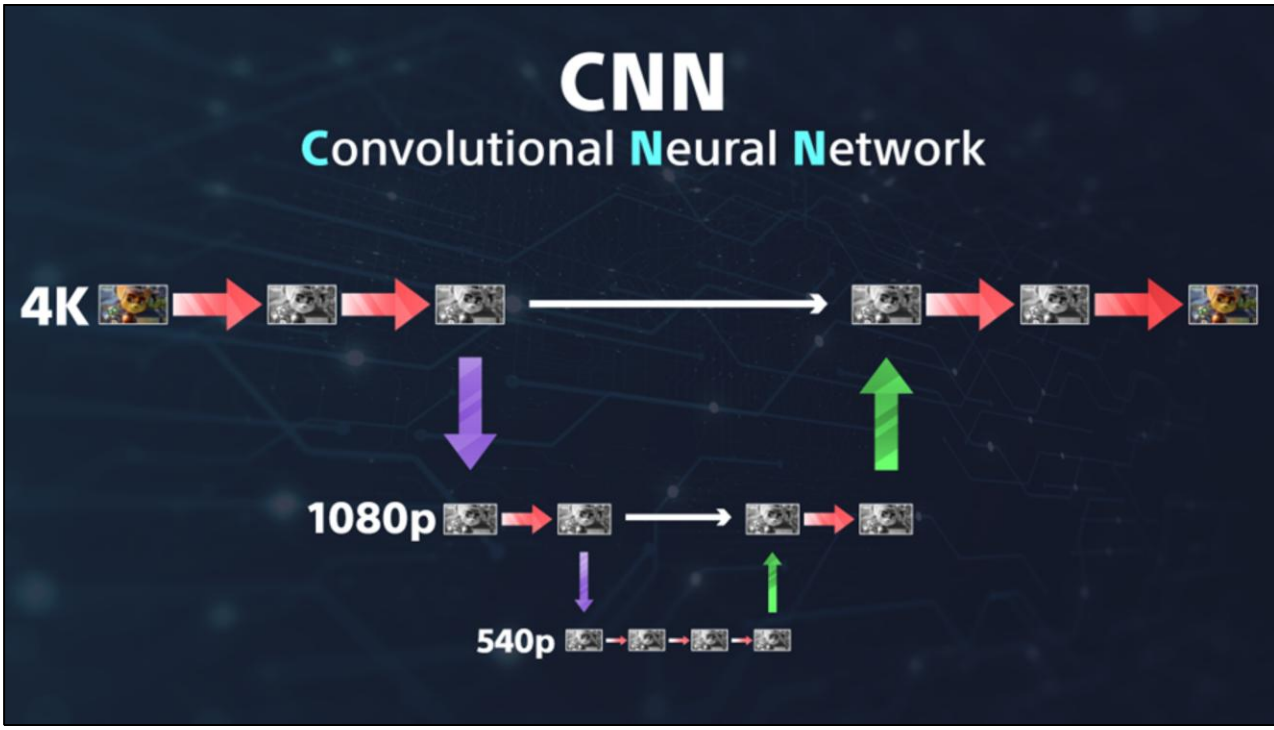
- A representative foliage example. It also shows the sharpness limitation covered next, which is clearer in the later comparison.

- A high-upscale detail example: stable and much improved over TAAU, with most detail present. Room remained for more clarity, plus rare, subtle color fringing at edges.

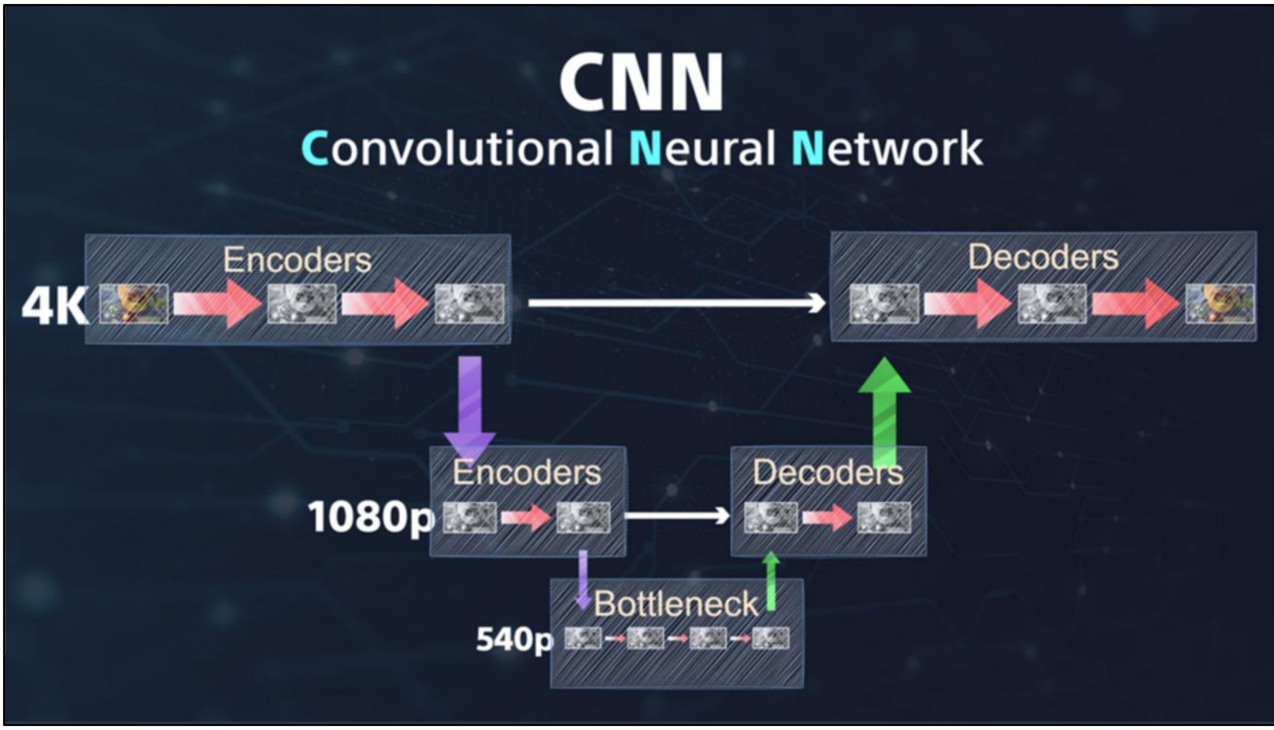
A Great Starting Point

- High-level architecture followed the latest research:
  - Color Predicting Network (CPN)
  - Convolutional Neural Network (CNN)
  - UNet-based architecture
- Algorithmic reprojection / resampling preceded the model
  - But most of the rest was done by the model

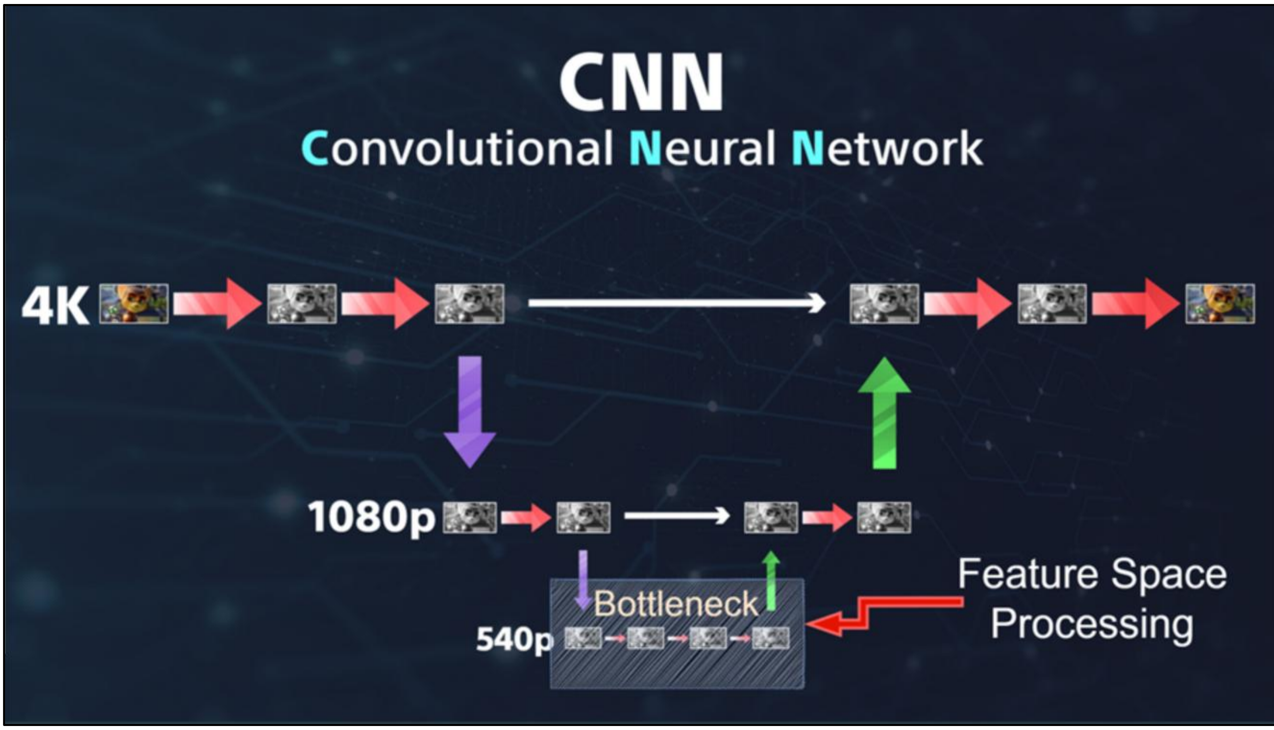
- For its time, this design was close to the research state of the art.



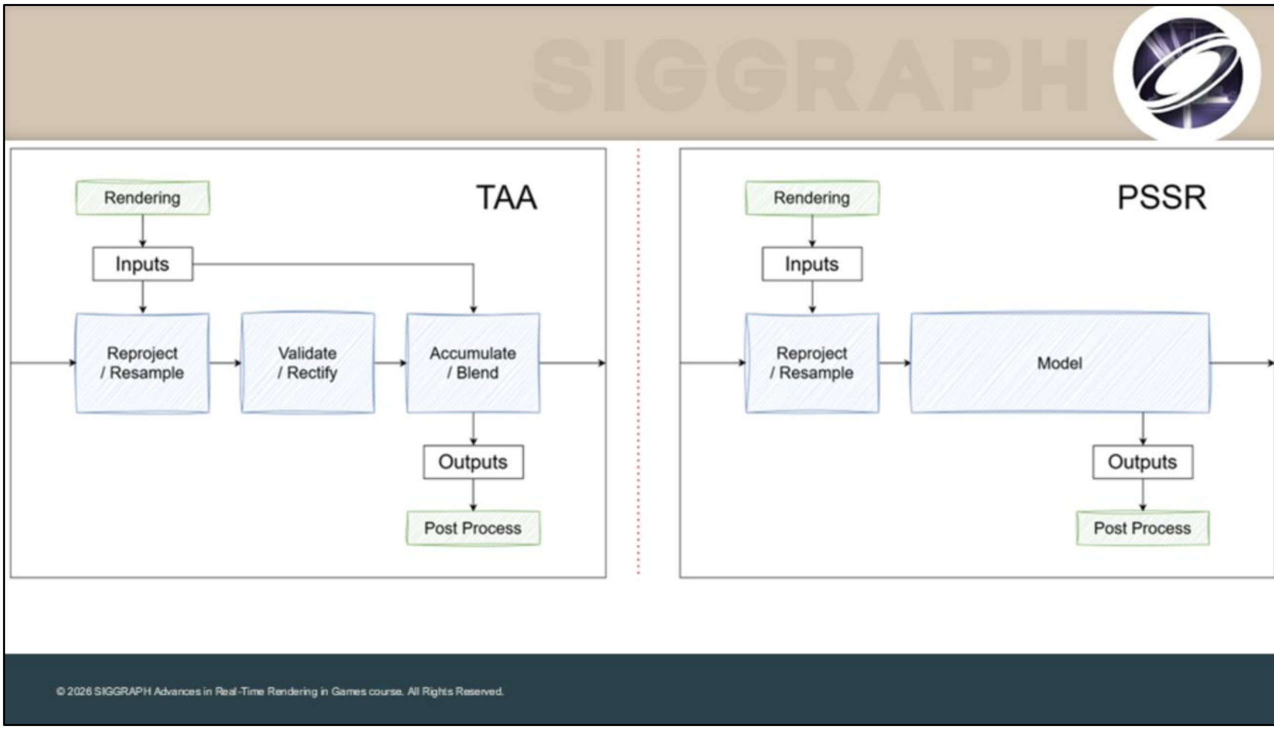
- Mark Cerny used this image in the original technical brief. The input frame is processed as a series of convolutions at multiple resolutions.



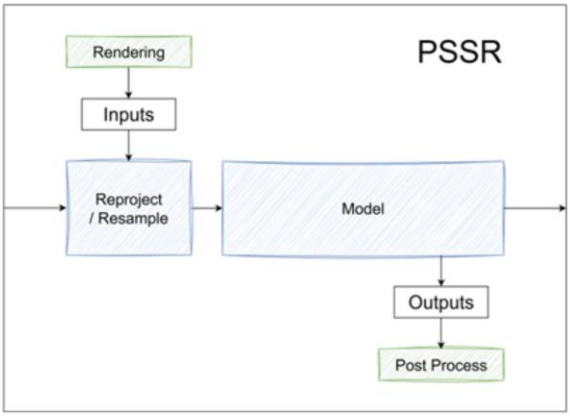
- Nomenclature: convolutions are matrix multiplies; a contiguous set at one resolution is a layer. Input-side layers are "encoders," the lowest is the "bottleneck," and output-side layers are "decoders."



- In ML terms: encoders create a latent feature space, the bottleneck processes within it, and decoders convert back.



- "Color Predicting Network" means the model outputs color directly. Relative to the earlier TAA breakdown, the model takes over the validate/rectify and accumulate/blend steps.



The architecture follows this initial intuition to

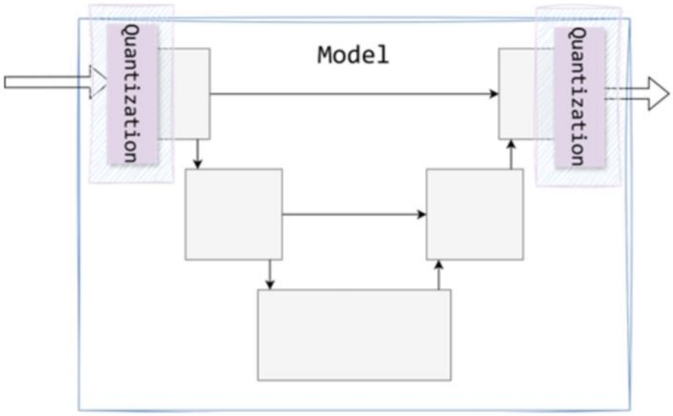
- Give the model as much responsibility as possible.
- Give the widest surface area possible to optimize against

Freedom Became Debt

- Quantization
- Storage / Recall
- Learning Classical Math / Filters
- Gradients

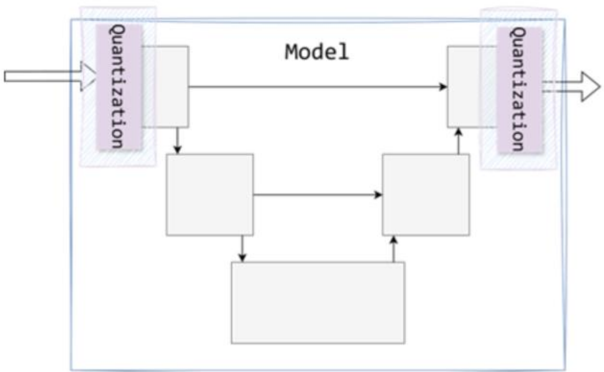
- In hindsight, giving a runtime-sized model such a wide optimization surface became a liability: several of these tasks are a poor fit for the model's instruction set. Each is covered next.

Quantization



- Quantization and dequantization are a real, expected cost.

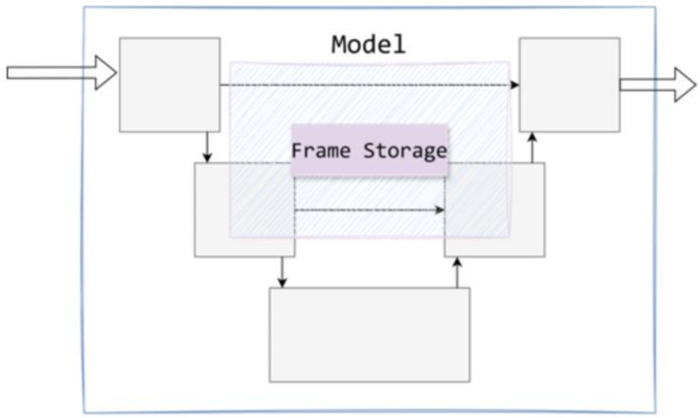
Quantization



- Input / Output is in 11:11:10 float
- But the model is 8-bit

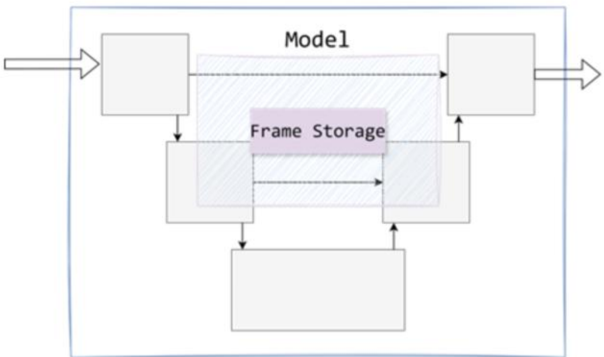
- Input and output are 11:11:10 float while the model runs 8-bit, so the model must learn a multi-channel encoding for HDR data. This also forced most training to run quantized and produced the rare color fringing shown earlier.

Storage / Recall



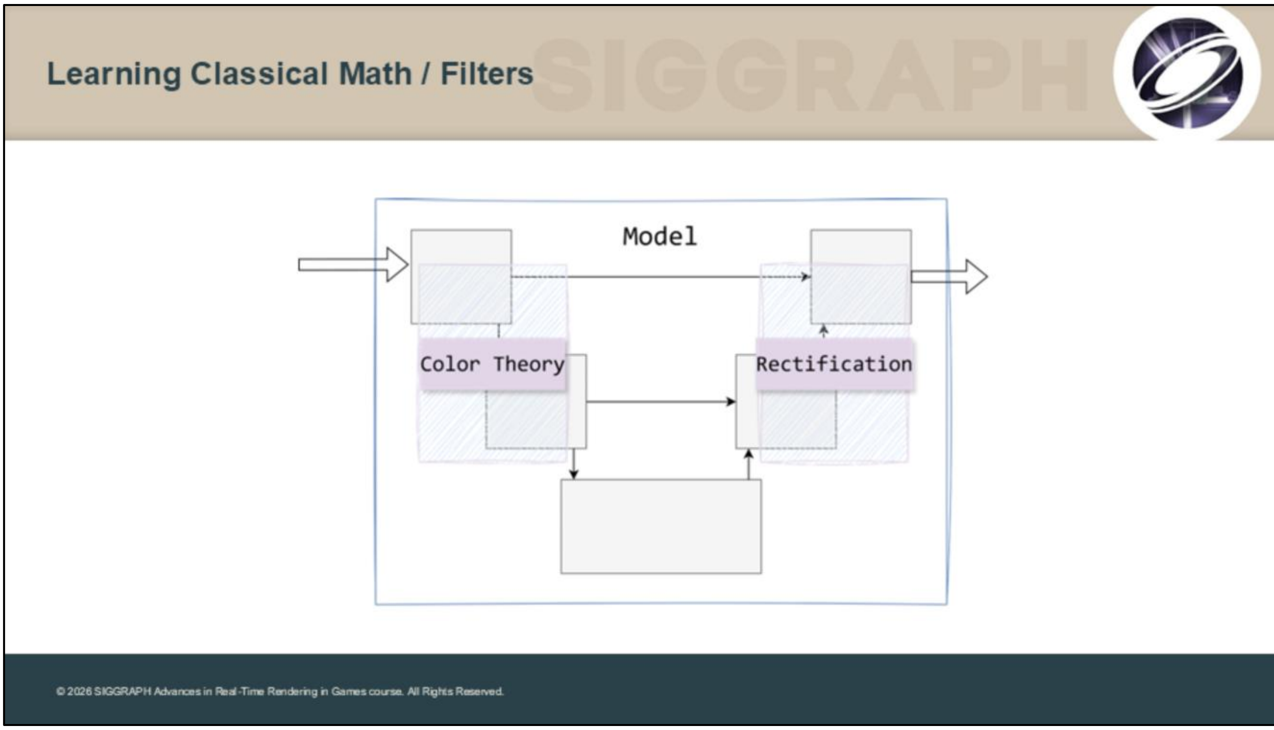
- The next fundamental issue is storage.

Storage / Recall



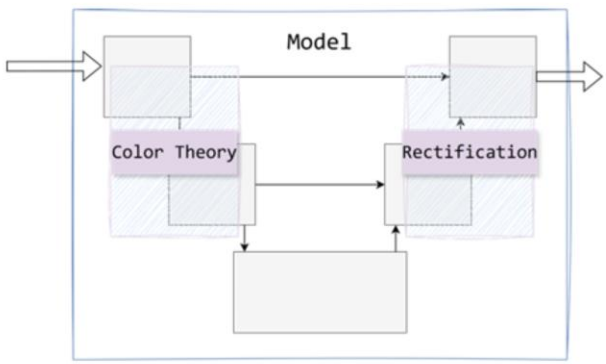
- Where do frame buffers go in between?
- While it's doing the real work?
- Model must implicitly store and reproduce HDR color
- (in 8-bit registers)

- The model must implicitly store and reproduce HDR frame buffers in 8-bit registers while it works. Whether via registers or an invertible encoding, this is an inefficient use of resources.



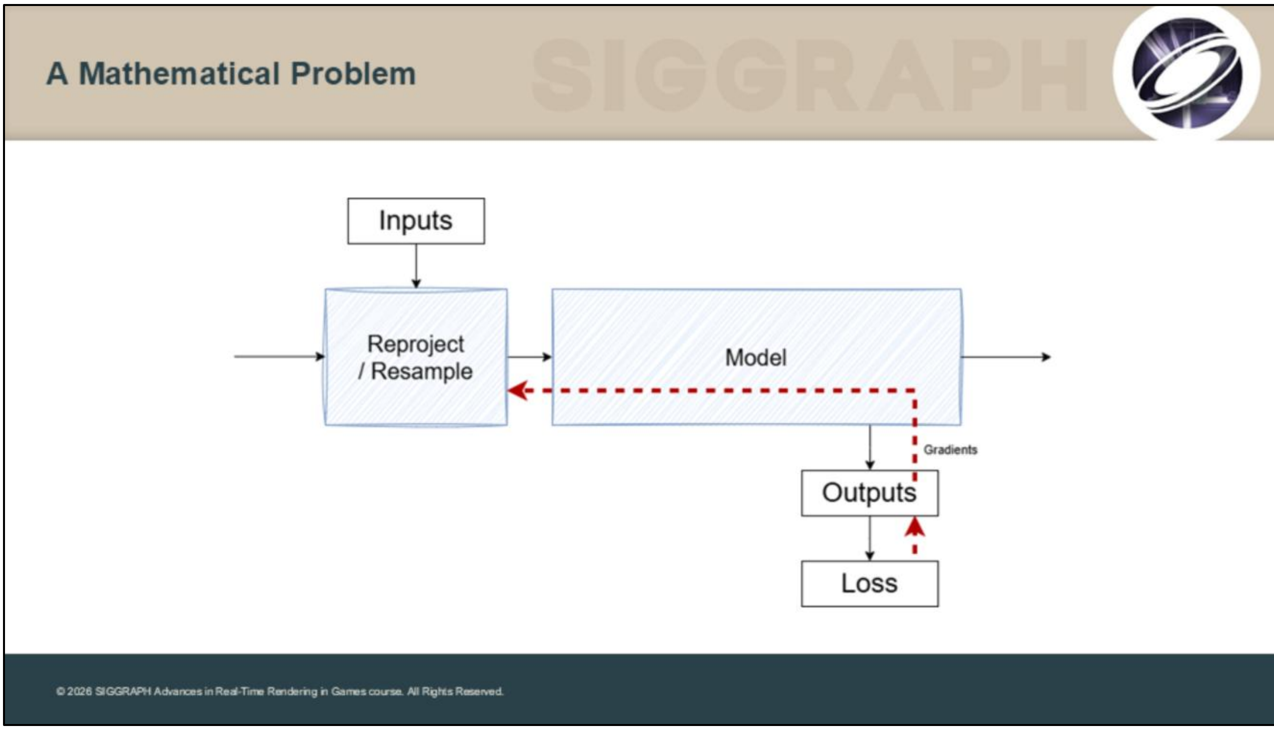
- The model also has to perform substantial shader-style math.

Learning Classical Math / Filters



- Model needs to learn much of the math involved in TAA
  - History rejection
  - History rectification
  - Temporal blending
  - Anti-aliasing
  - Firefly rejection
- Model must implicitly learn these
  - With a very limited instruction set

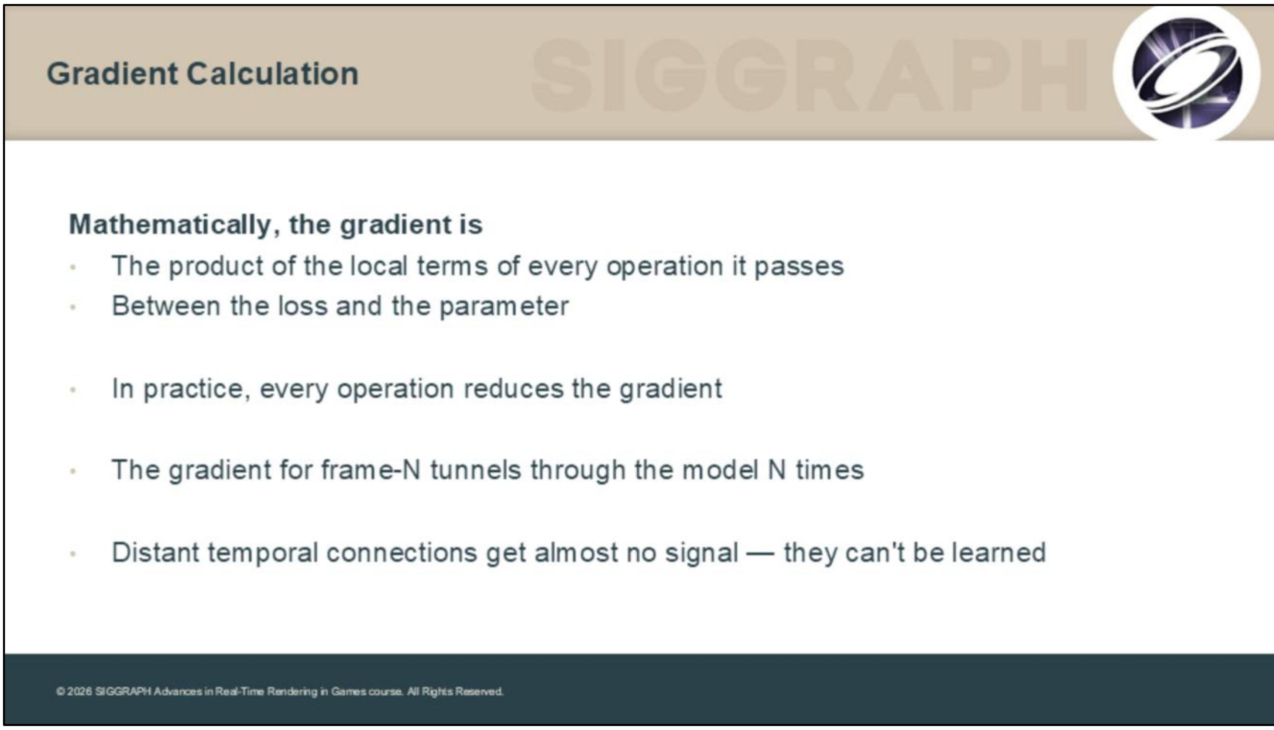
- The model must implicitly learn the math of TAA's accumulate/blend step (history rejection, rectification, blending, anti-aliasing, firefly rejection) with a limited instruction set. It can, but whether that is efficient was worth testing.



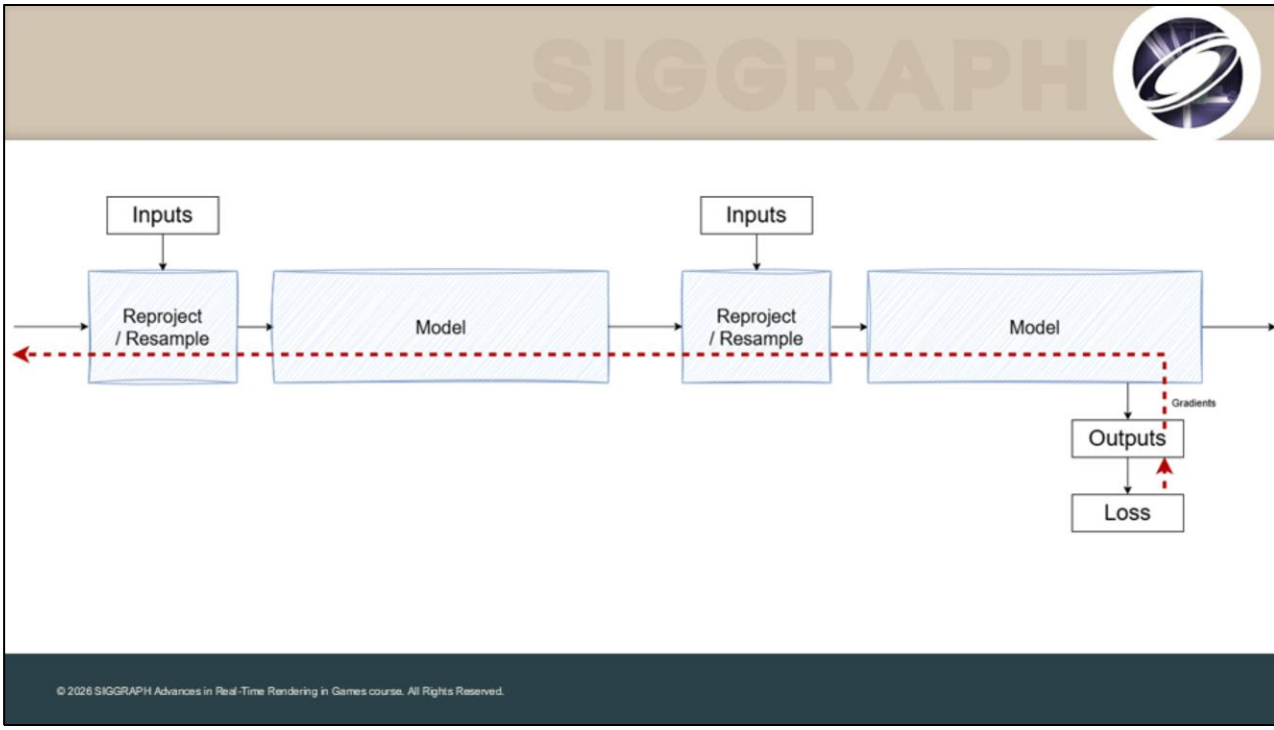
- A further challenge was purely mathematical, affecting both the model and the trainer.

How the Model Learns

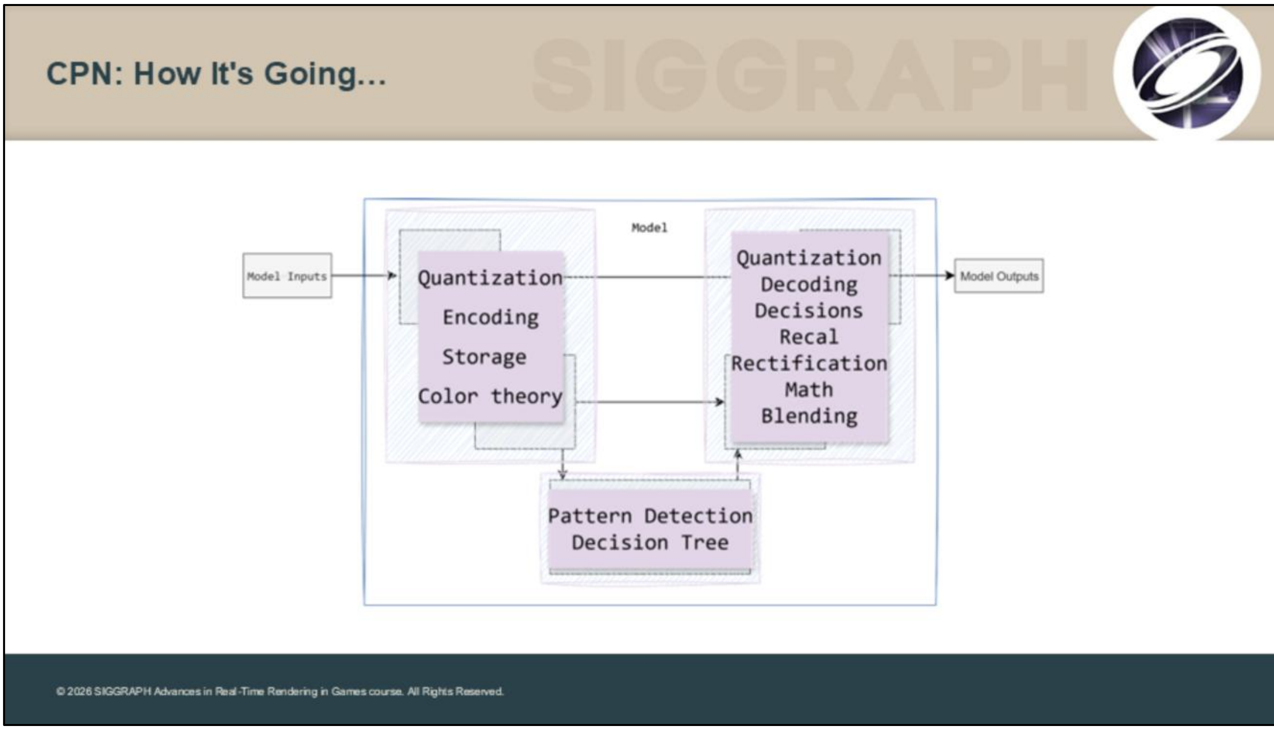
- The model parameters define
  - The convolution filters (the patterns it looks for)
  - The combination weights (the decisions it makes)
- Training minimizes a single loss value, L
  - By optimizing the parameters
- Training nudges each parameter based on its gradient
  - How much it moves the loss



- The gradient is a product of local terms, one per operation on the path from loss to parameter, so every operation scales it down. This "vanishing gradient" effect accumulates, biasing the model toward shorter associations. Residual and skip connections reduce but do not eliminate it.



- As a result, distant temporal connections were very hard to learn regardless of model size. Launch PSSR handled a handful of frames well, better than TAAU, but struggled much beyond that.



- With a lot happening inside the model, the goal for the next version was to simplify it against clear quality targets and a clear technical plan.

- This section opens with a brief PC comparison for context.

SR Continues

- On PC, ML-based upscaling / reconstruction was advancing
  - By PSSR launch PC solutions were much more mature
  - In many cases improving games in place
  - Which is great!

- During Launch PSSR development, PC ML upscaling matured and increasingly improved games in place. On console the relevant framing is quality per cost.

Requirements

To succeed we needed:

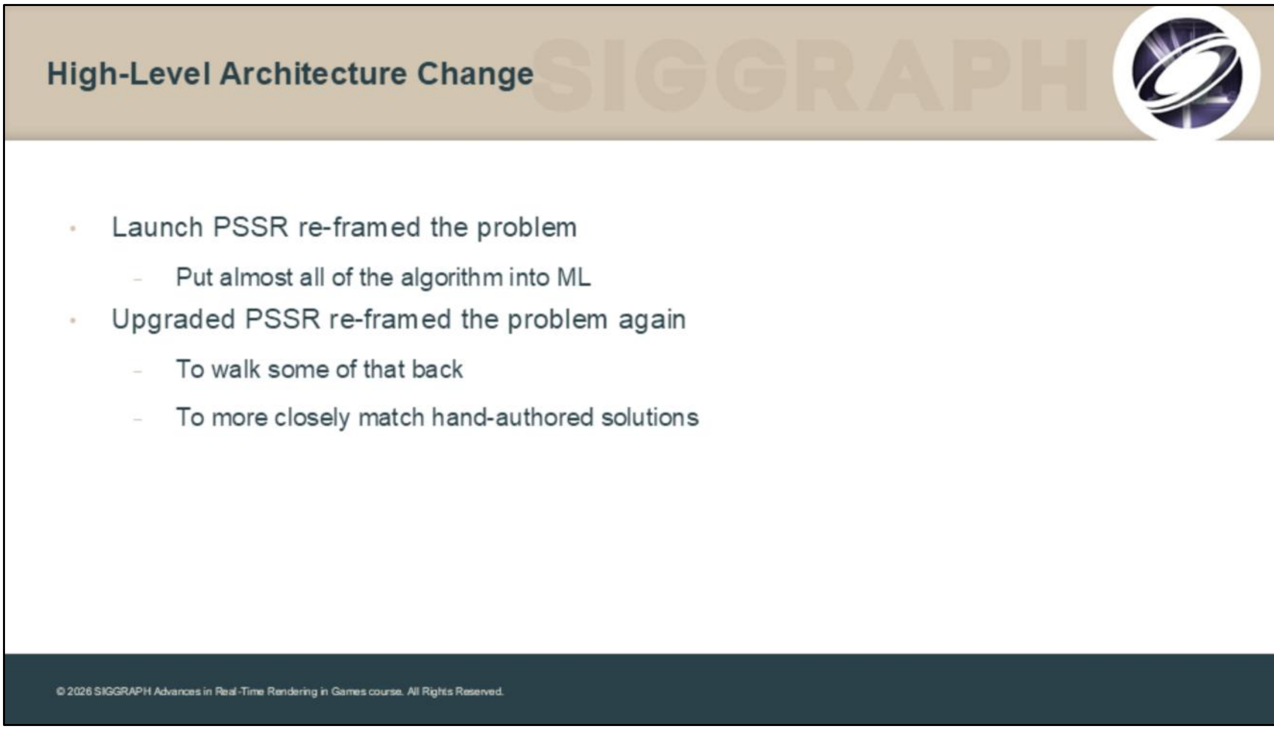
- Prominent / Conspicuous quality improvements
- No (Prominent / Conspicuous) quality regressions
- No performance regressions
- No memory regressions

- On PC quality had scaled, but costs also scaled. This would not work for us.

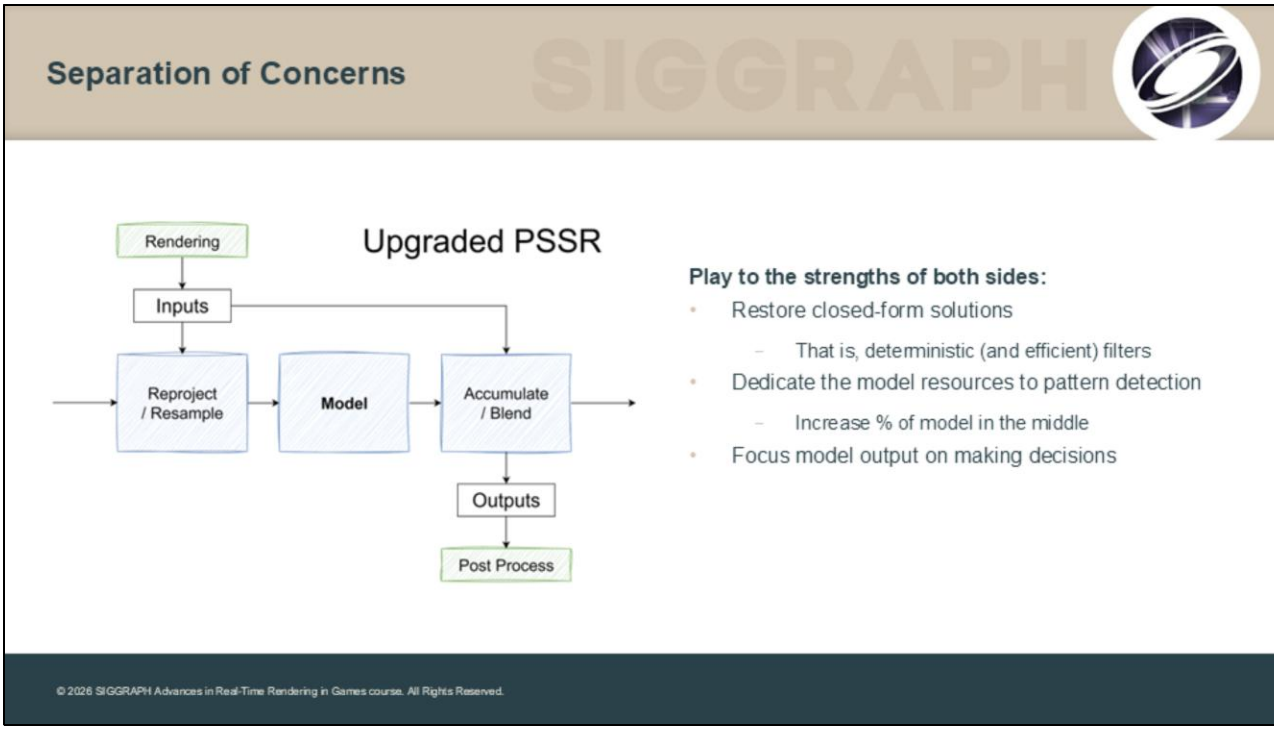
Upgraded PSSR: Changed Everything

- High-level architecture
- Internal block architecture
- Training data and augmentation
- Training system optimization

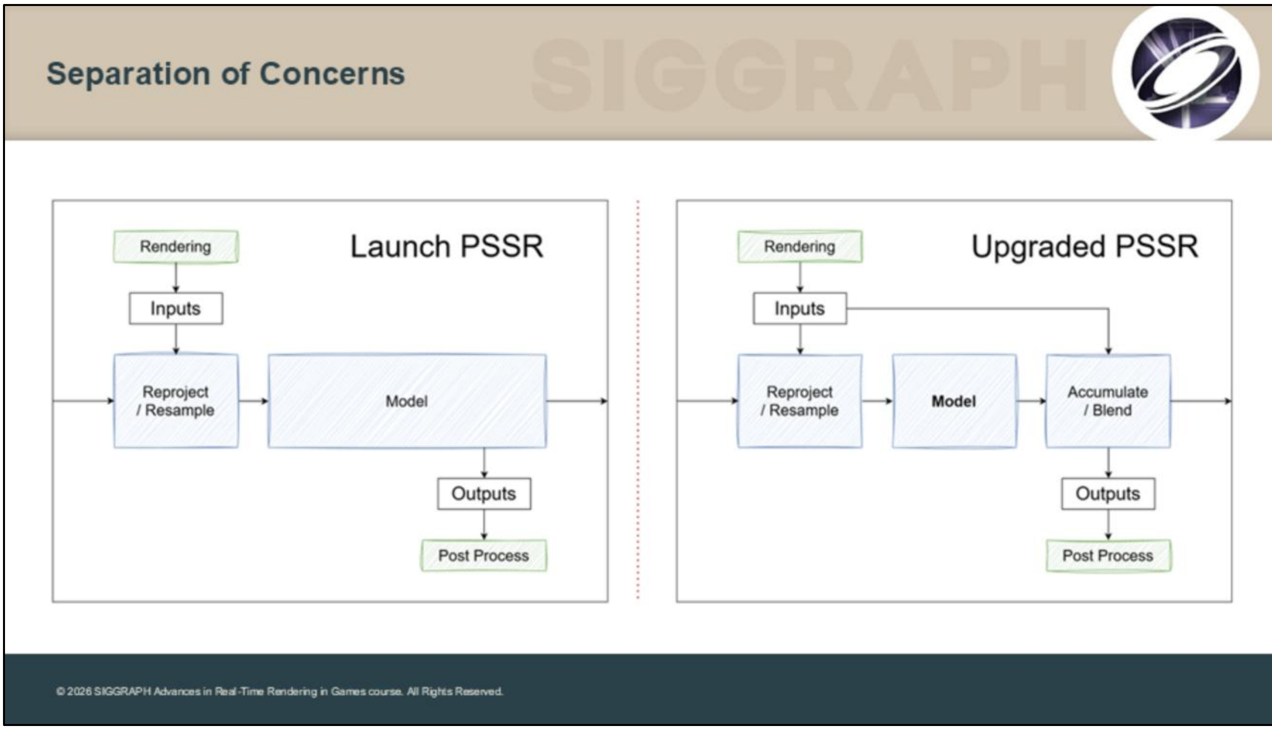
- Every part changed: design, implementation, training, and optimization. This talk focuses on the high-level architecture, the most fundamental and impactful of these.



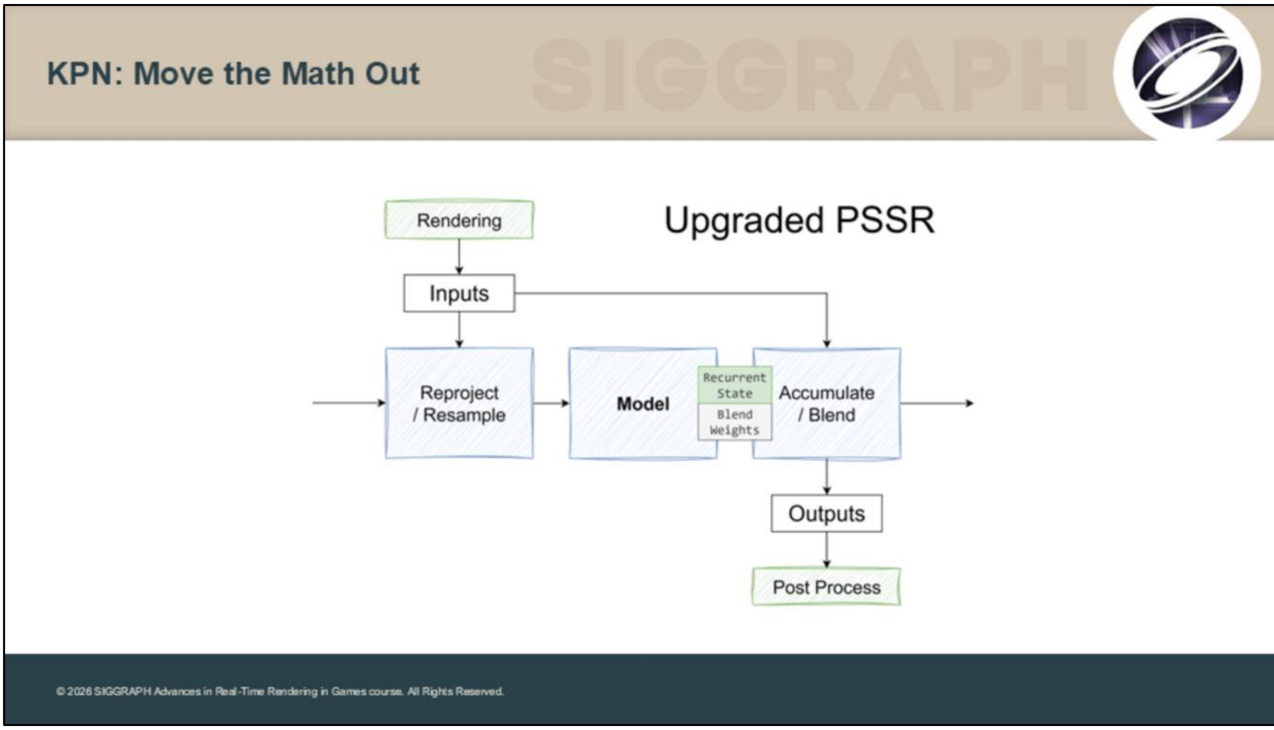
- Where Launch PSSR moved almost the entire algorithm into ML, the upgrade walks part of that back, using closed-form solutions where possible and giving the model more focus on pattern detection and decisions.



- Closed-form solutions exist for many of these steps, and the GPU instruction set runs them efficiently; the ML instruction set is built for pattern detection. The approach uses each for its strengths.

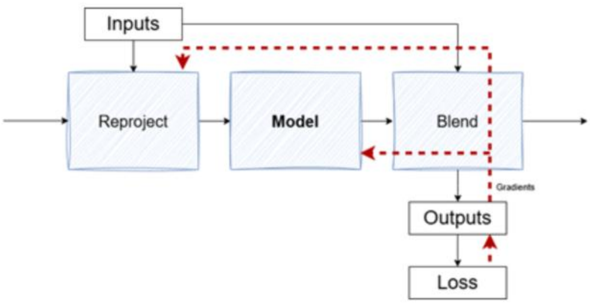


- This largely restores TAA's three logical blocks, with the model focused on the heuristics in the middle.



- The CPN was replaced with a Kernel Predicting Network (KPN): the model outputs blend weights rather than color, driving the filtering while the blend is applied explicitly. As a result the model no longer stores or blends HDR color in 8-bit; blend weights fit naturally in 8-bit, and encoder/decoder blocks spend less effort on overhead and more on the intended work.

Gradient Flow

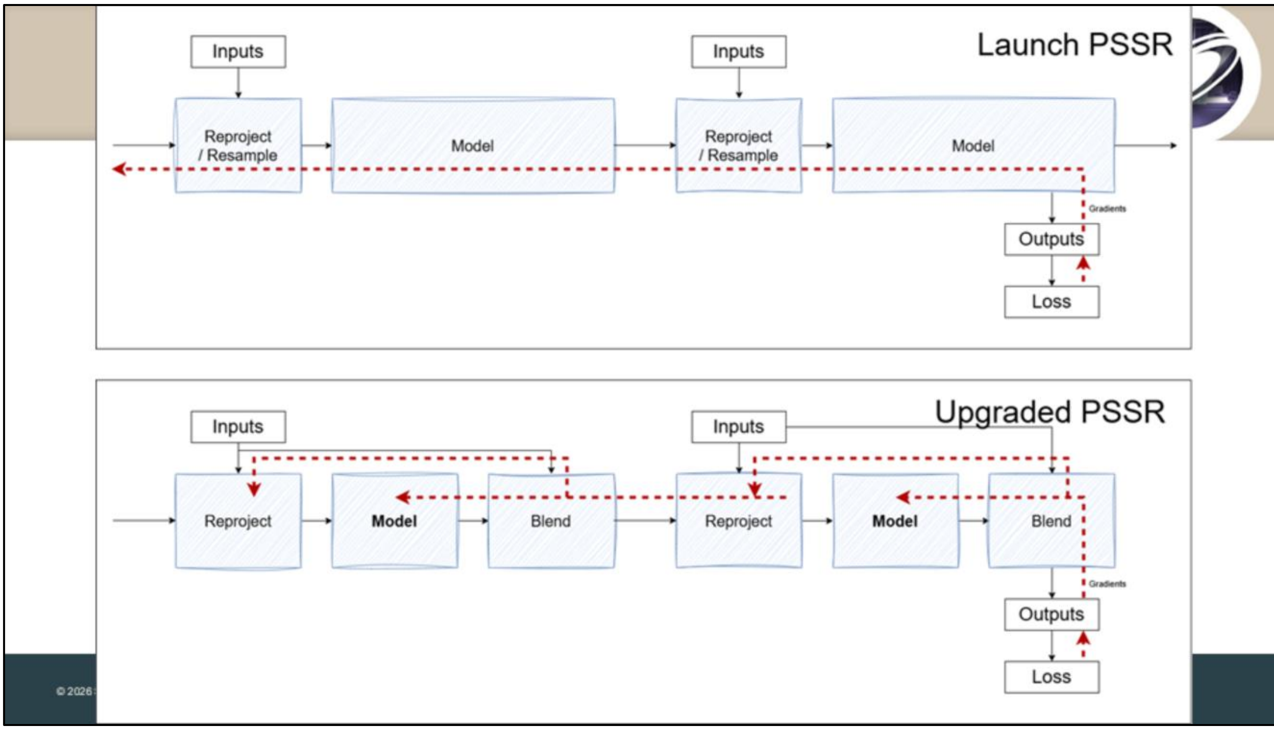


- KPN outputs blend weights
- Gradient to history color is then a single multiply
- The signal flows *around* the model
- Straight to reprojection
- In ML terms this is considered a GRU
  - (GRU = Gated Recurrent Unit) [CMG*14]

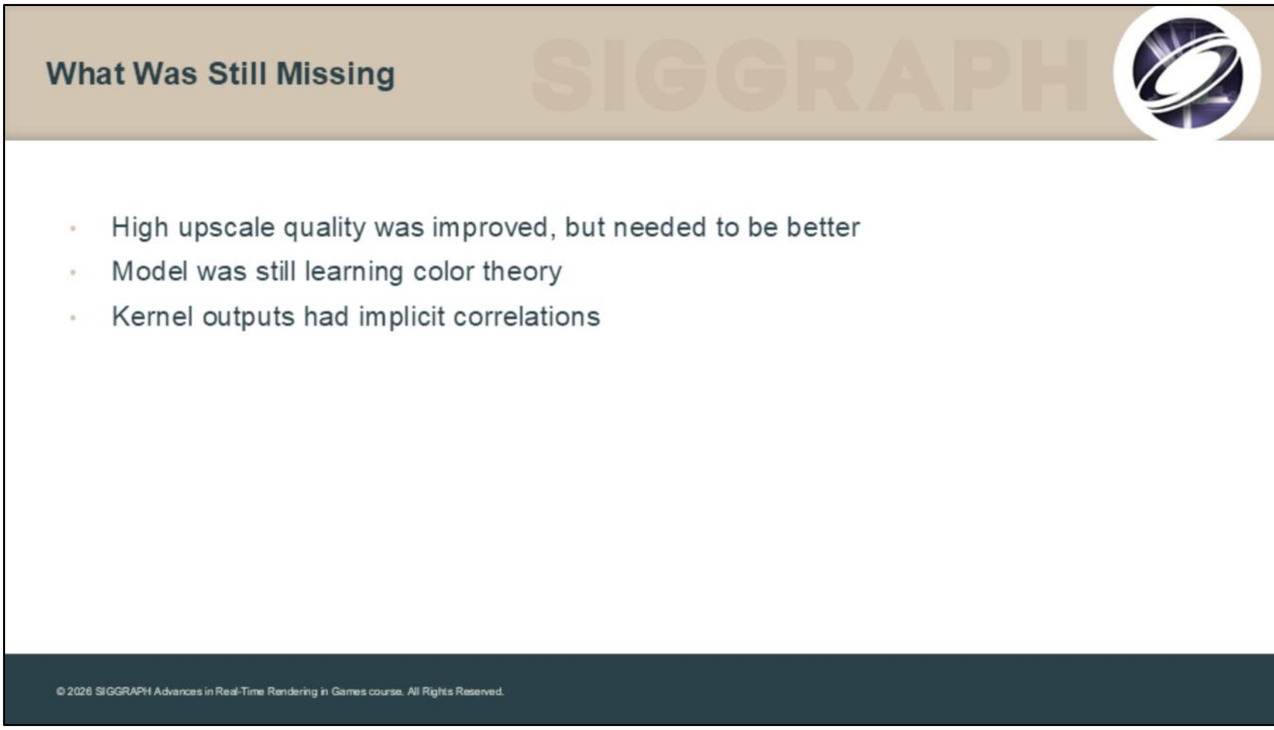
Now:

- Distant temporal associations much easier to learn
- Temporal feedback is more effective

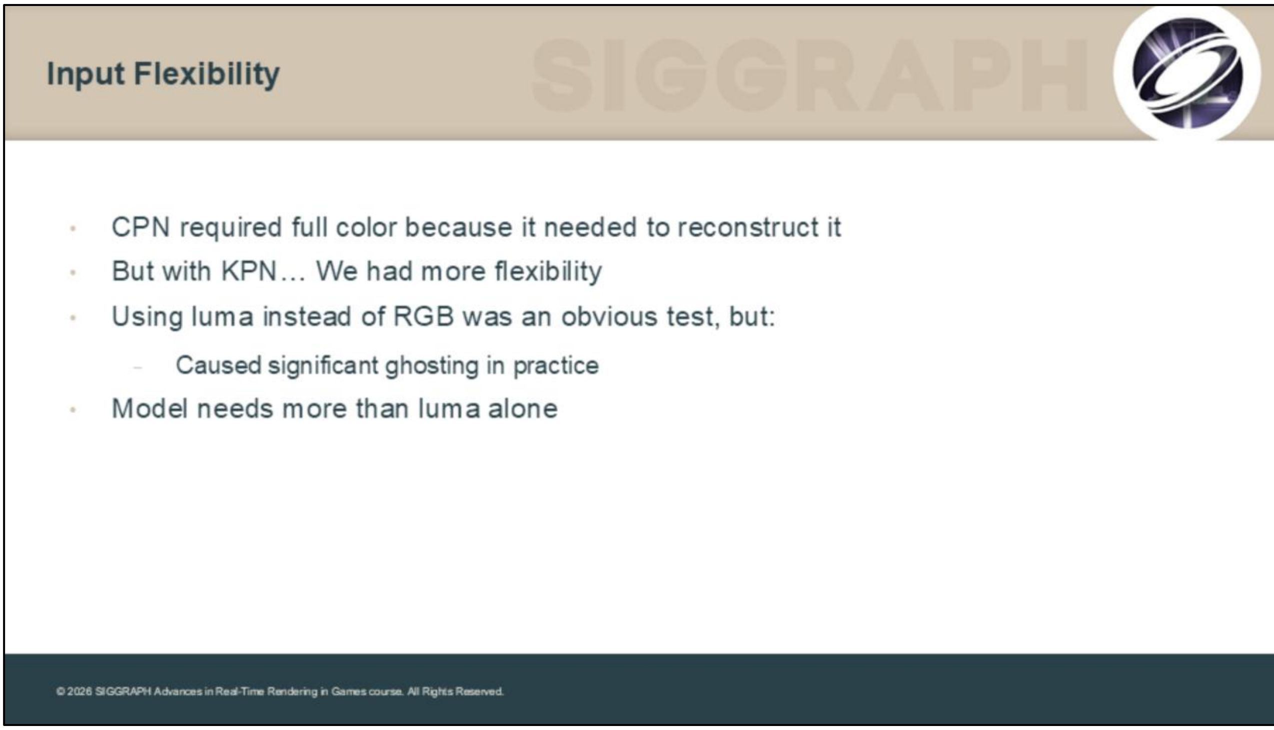
- Adding the explicit blend and reconnecting it to the inputs fixed the gradient-flow problem: with a weighted sum, the gradient to history color is a single multiply. This gating structure is a GRU.



- The learning signal now routes around the model directly to reprojection and the prior frame instead of tunneling through it serially, the mathematical fix that enabled much more distant temporal connections.



- Progress from a small change, but more was wanted: further reduce implicit math on inputs and outputs, improve input color handling, and improve detail at higher upscale ratios (2.5x, 3x).



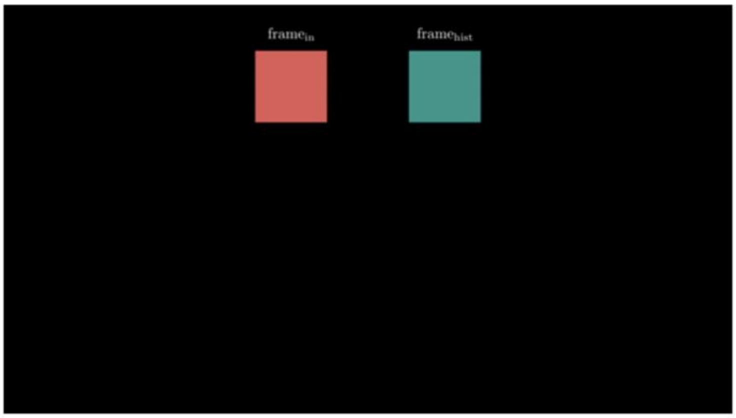
- The CPN needed full color to reconstruct it; the KPN's blend-weight outputs allow more flexible inputs. Luma alone was an obvious test but caused significant ghosting, so the model needs more than luma.

Make Color Relative

- Convert RGB → YCbCr (input + history)
- Pass luma (Y_in, Y_hist) directly
- Pass chroma distance (ΔCbCr) instead of raw chroma

- Color is made relative rather than absolute via a "color distance": luma passed directly, chroma passed as a distance (delta CbCr). The simple option proved effective.

Color Distance



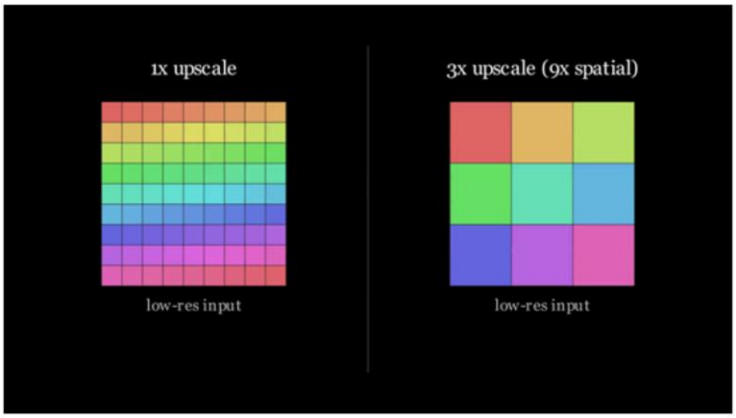
- Color distance is the Euclidean distance between the input and history Cb,Cr channels. The model becomes aware of color differences without needing absolute color, which trains well, correlates with outcomes, and generalizes, while avoiding absolute-color biases from the training data (e.g. "particles are orange," "sky is blue").

Output Issues

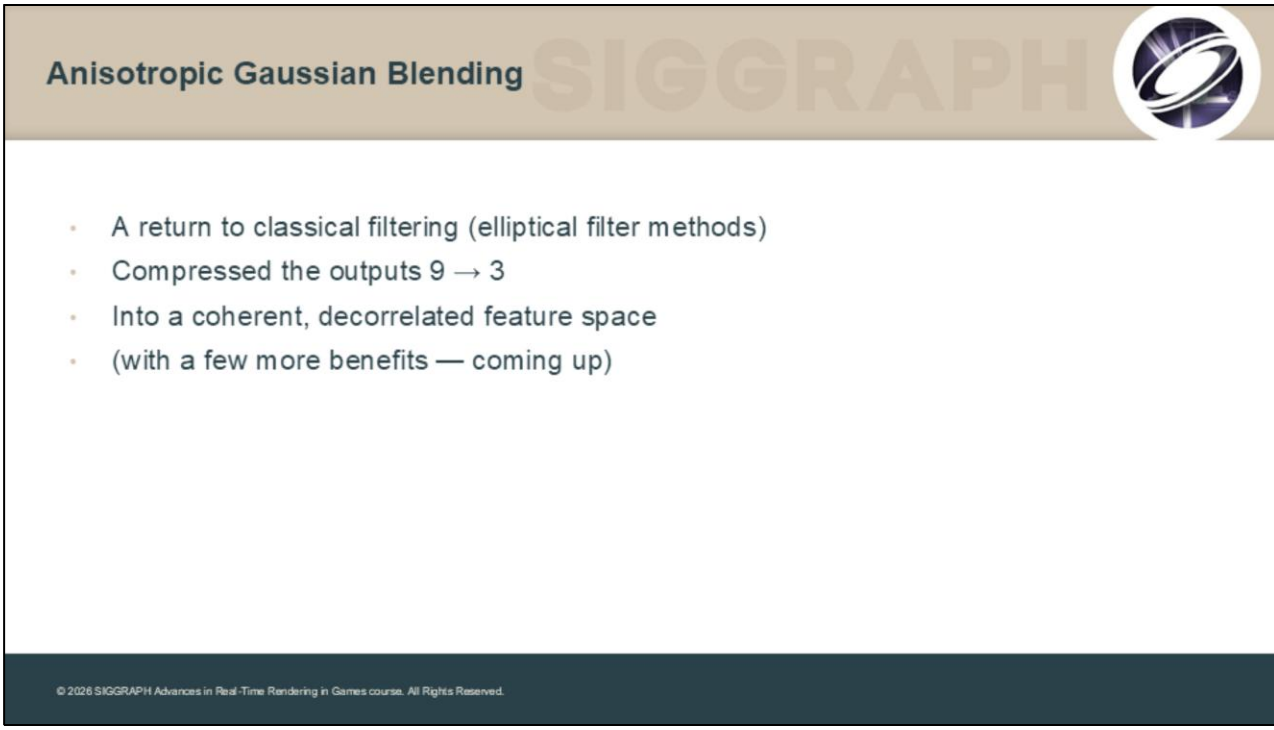
- 3×3 kernel outputs were highly correlated
- Didn't generalize well at high upscale ratios
- Could they be more concise and scale with input?

- The 9 outputs of the 3x3 kernel trained acceptably and improved on the CPN, but were highly inter-correlated and did not generalize well at high upscale, suggesting room for a more concise, input-scaling representation.

Kernel Reach

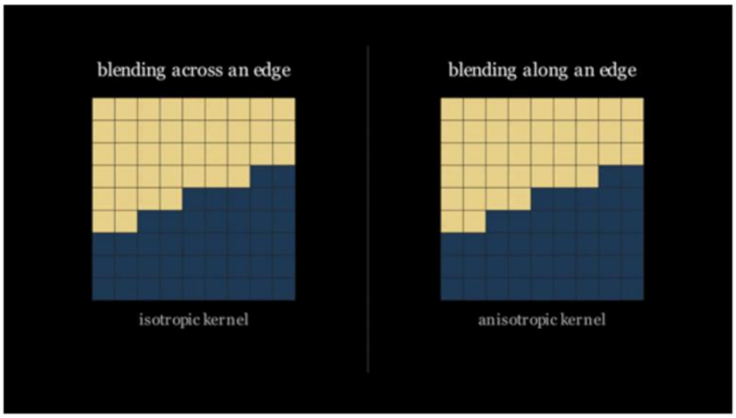


- The canonical kernel is applied at output resolution, so its reach in input pixels shrinks as the upscale ratio rises. At 3x it provides almost no blending.



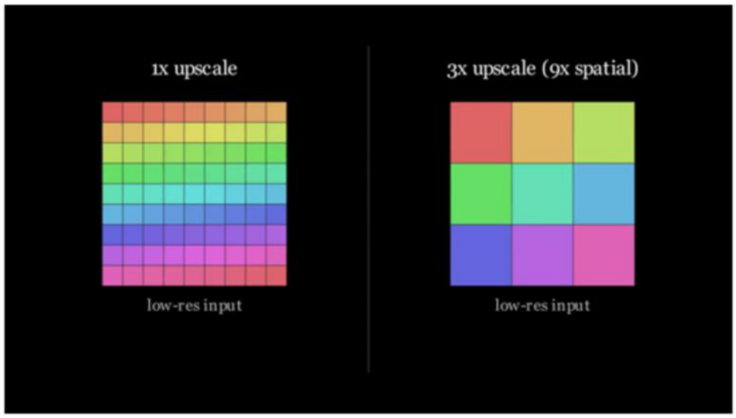
- The chosen solution returns to classical elliptical filtering with an anisotropic Gaussian, compressing the 9 outputs to 3 and decorrelating them, with further benefits following.

Blending Along Edges



- Elliptical blending lets the model blend along oriented edges, effective for building and retaining detail, driven by local feature strength and orientation.

Blending in Input Space



- The kernel is also scale-invariant. Applied in input space, it always has 9 distinct input samples and performs real blending whether at 1x or 3x, with reach that scales with input resolution, a significant gain in high-upscale detail.

Preserving Artistic Intent

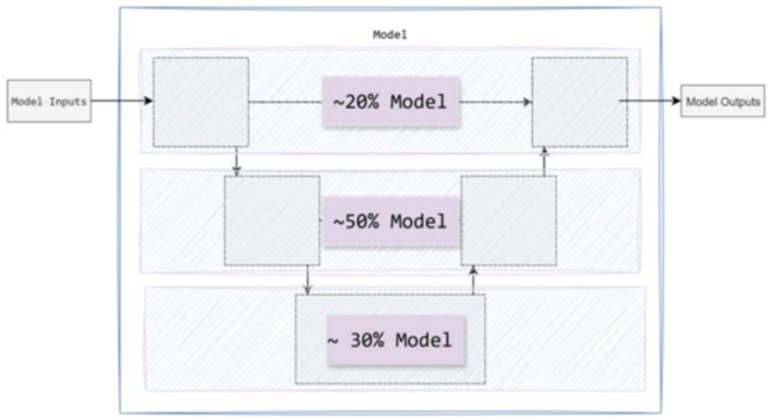
- Another benefit is tied to preserving artistic intent
  - Kernel prediction practically prevents hallucination
  - Output is confined within the convex hull of input samples
  - Hence the "reconstruction" nomenclature
- Early on we worried that limiting the model would degrade quality
  - Instead it has consistently outperformed expectations
  - And our previous model

- Kernel prediction structurally limits content changes: the Gaussian cannot exceed the convex hull of the input samples, so the model is demonstrably reconstructive rather than generative, which is useful when addressing teams' concerns. Early worries about a quality cost did not materialize; results were consistently positive.

Output Impact

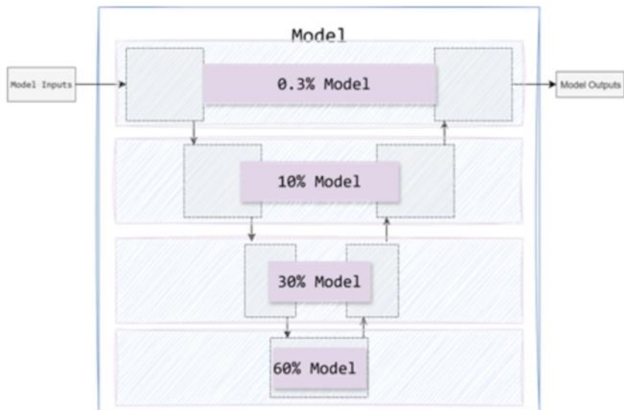
- Quality jumped up for high upscale ratios
- Model was more efficient

CPN/Before:



- Weight distribution of the CPN versus the intended design: much of the capacity sits in the upper (high-resolution) levels, leaving only ~30% of parameters at the lowest level (540p).

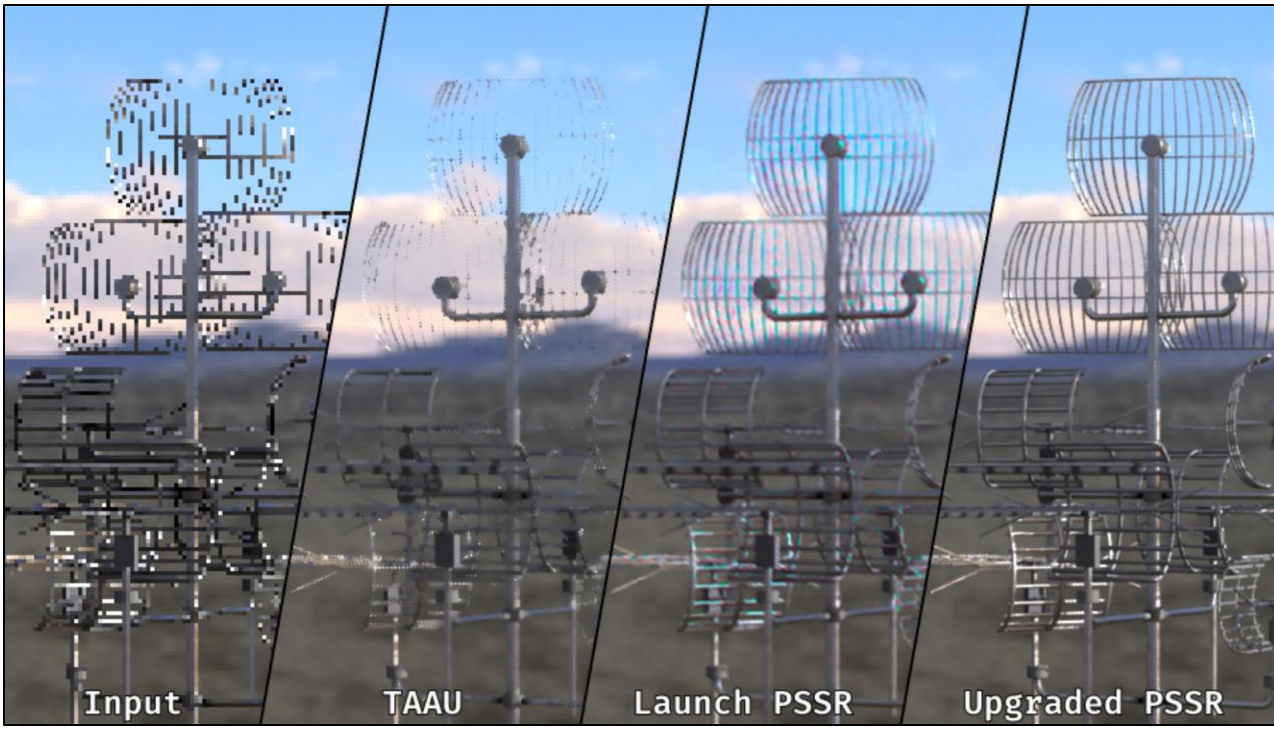
KPN/After:



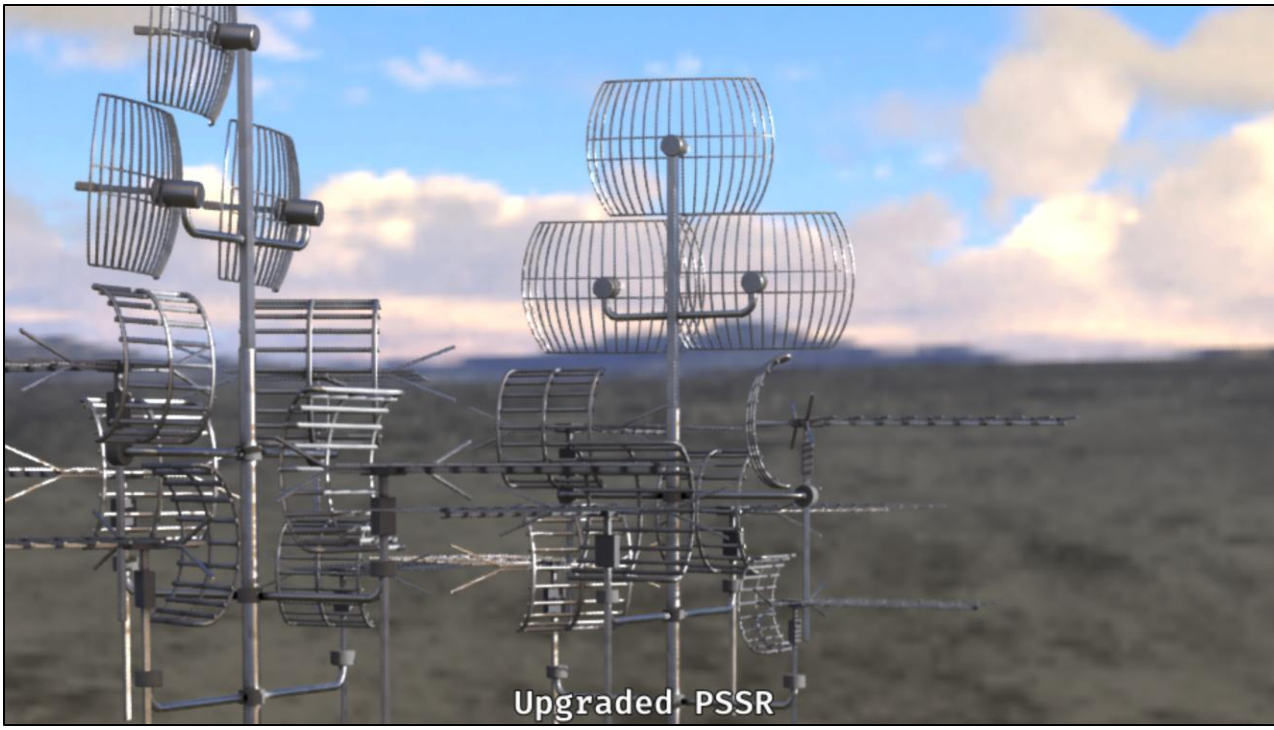
- The KPN does very little at 4K and adds a new lower layer, placing ~90% of parameters at 540p or below, matching the intended distribution toward lower levels.

- Returning to several earlier images, plus new ones.

- The same charging scene on Upgraded, showing detail recovered in the high-frequency regions that broke up under TAAU.

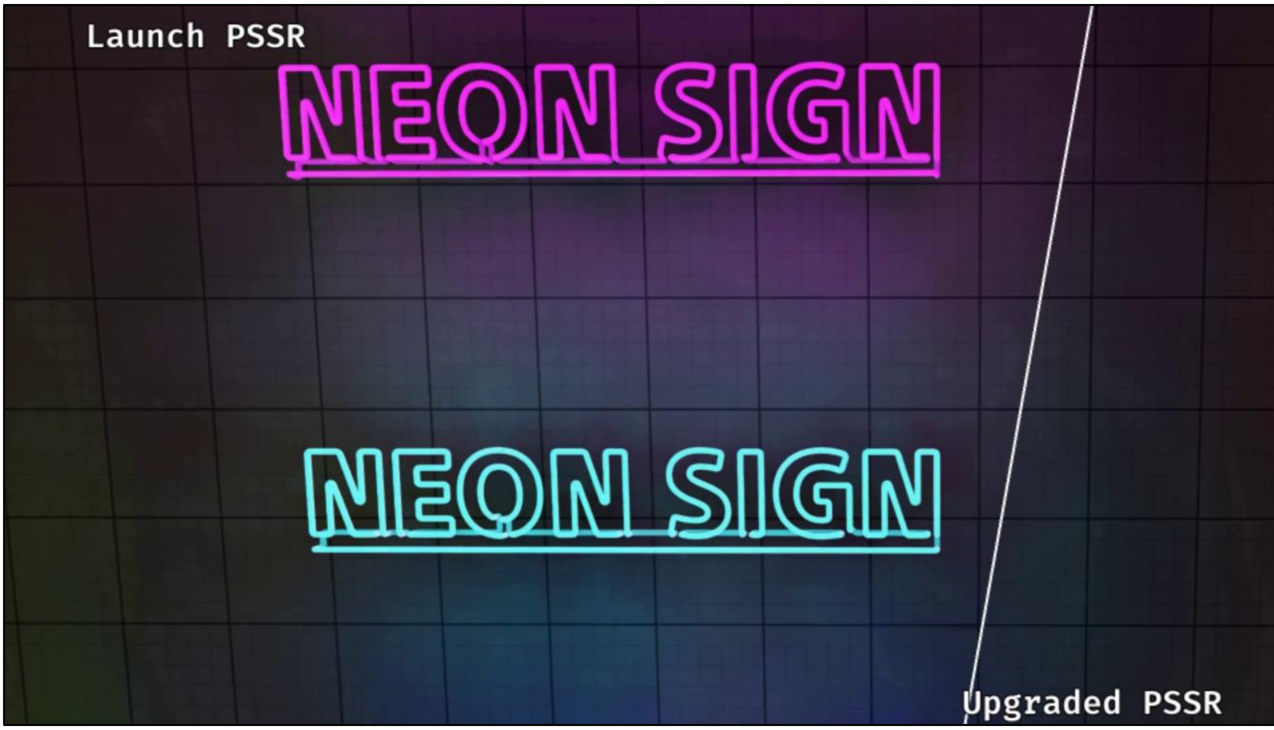


- The antenna zoom from the opening: sharper, with the fringing gone.



- The full frame: stable, with detail holding together.

- Foliage with lighting flicker: sharpness and detail hold up, and the flicker settles relative to Launch.



- The neon comparison again: pulsing on the left (Launch), stable on the right (Upgraded).

Results

- Improved:
  - Image quality
  - Temporal stability
  - Lower memory + runtime cost
- Training: ~4 GPU months → <1 GPU week

- Higher quality at all input resolutions and notably at lower ones, reduced temporal instability, and lower GPU time and memory. Training time dropped from ~4 GPU months to under a GPU week, largely from the quantization changes, enabling much faster iteration.

Acknowledgements

Project Amethyst SR Research Team
Sony PSSR Research Team
Sony PSSR SDK / Production Team
Mark Cerny
Luis Villegas
Shigetaka Kudo
Rui Yamada
Geoff Norton
Maurizio Cerrato
Natalya Tatarchuk
Marta Orlinska
Benjamin Coquelle
Maria Chiara Monti
Yuji Yamada
Yiannis Andreopoulos
Aaron Chadha
Sebastian Lutz
Paulwei Wang
David Simpson
Kleber Garcia

- Acknowledgements span the research teams (AMD / Project Amethyst and Sony's PSSR researchers), the SDK and production teams, and those who reviewed and approved the deck. Research is a small part of what it takes to ship.

SIGGRAPH 2026
Los Angeles 19–23 JUL

Questions

References

- **[Lot11]** FXAA — Timothy Lottes — 2011 - NVIDIA — https://developer.download.nvidia.com/assets/gamedev/files/sdk/11/FXAA_WhitePaper.pdf
- **[JME*11]** Practical Morphological AntiAliasing. — JIMENEZ J., MASIA B., ECHEVARRIA J. I., NAVARRO F., GUTIERREZ D. — 2011 - GPU Pro 2
- **[JES*12]** SMAA: Enhanced Subpixel Morphological Antialiasing — Jorge Jimenez, Jose I. Echevarria, Tiago Sousa, Diego Gutierrez — 2012 - EUROGRAPHICS Volume 31, Number 2

References

- **[LG21]** Improved Spatial Upscaling through FidelityFX Super Resolution for Real-Time Game Engines — Timothy Lottes, Kleber Garcia — 2021 - Advances in Real-Time Rendering in Games, SIGGRAPH
- **[Kar14]** High-Quality Temporal Supersampling — Brian Karis — 2014 - Advances in Real-Time Rendering in Games, SIGGRAPH
- **[Wih17]** 4K Checkerboard in Battlefield 1 and Mass Effect — Graham Wihlidal — 2017 - Game Developers Conference — https://www.ea.com/frostbite/news/4k-checkerboard-in-battlefield-1-and-mass-effect-andromeda

References

- **[Aba21]** Temporal Super Resolution in Unreal Engine 5 — Guillaume Abadie — 2021 - Advances in Real-Time Rendering in Games, SIGGRAPH
- **[YLS*20]** A Survey of Temporal Antialiasing Techniques — Lei Yang, Shiqiu Liu, and Marco Salvi — 2020 - EUROGRAPHICS Volume 39, Number 2 — http://behindthepixels.io/assets/files/TemporalAA.pdf
- **[CMG*14]** Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation. — Kyunghyun Cho, Bart van Merriënboer, Çağlar Gülçehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. — 2014 - EMNLP — https://arxiv.org/pdf/1406.1078