Adaptive Quality Map Hyperstacking for Pixel-Wise Quality-Weighted Deep-Sky Reconstruction

Method definition v0.2.0, implementation contract, post-reconstruction robustness extensions, and an M31 validation run

Jeamy Lee[1]

[1]Independent Researcher / tile_compile project

July 22, 2026

Abstract

Adaptive Quality Map Hyperstacking (AQMH) is a deterministic pixel-wise quality-weighted reconstruction method for short-exposure deep-sky imaging. It replaces block-constant local quality scores with dense per-frame quality maps and reconstructs each supported output pixel by a weighted mean whose AQMH weight is evaluated at that same pixel. The method is independent of Classic Tile Compile: registration, photometric normalization, common-overlap handling, reporting, and run management may be shared, but AQMH does not use classic local tile metrics, state clustering, or synthetic frames as reconstruction inputs or fallbacks. This paper consolidates the AQMH v0.2.0 definition, including the post-reconstruction robustness extensions formalised in the v0.2.1 methodology: a background-gradient penalty in global frame quality, a registration-weight guard, adaptive low-frequency neutralisation, and structure-masked detail blending. It states the implementation contract used by tile_compile and evaluates a concrete M31 validation run. The run processed 645 frames on a 3924×2310 canvas, wrote 645 dense luminance quality maps at half resolution, and reconstructed the image with 5.350×10^9 finite map samples, zero missing map samples, and no fallback to classic tile weights. The measured median FWHM changed from 12.329 px to 5.268 px, corresponding to a 57.273% reduction. The background RMS changed from 4.185 to 0.780, corresponding to an 81.351% decrease. The uniform-control validation gate triggered on the background-RMS regression ($0.168 > 0.050$ threshold); the raw AQMH output was preserved by the guard. These measurements demonstrate a single-run execution of the AQMH path with the full v0.2.0 extension suite active; they do not by themselves establish population-level performance across object classes, instruments, or atmospheric regimes.

Keywords: astronomical image reconstruction, deep-sky imaging, quality maps, pixel-wise weighted mean, short exposure stacking, deterministic reconstruction, sigma clipping, uniform control validation

1 Introduction

Short-exposure deep-sky imaging collects many noisy frames instead of a small number of long exposures. The data often contain useful signal even when individual frames suffer from local defects: cloud edges, satellite traces, registration remnants, small hot-pixel clusters, or spatially varying seeing. A single global frame score cannot describe these local variations. Coarse tile scores improve locality, but they still assign one scalar quality value to an extended region. This has two structural consequences. First, a small contaminated area can depress the weight of a much larger block. Second, neighboring blocks can receive different constant weights, creating a discontinuous weighting field that is unrelated to the optical image content.

AQMH is documented here as an independent reconstruction path within the tile_compile project. The reference method for Classic Tile Compile is *Tile-Based Quality Reconstruction for DSO*, Version 3.3.9.

AQMH addresses the locality problem by computing a dense quality field $Q^{\mathrm{map}}_{f,c}(x, y)$ for each frame f and channel or stream c. The reconstruction then uses the frame sample itself and the AQMH weight field at the same pixel:

$$R_c(p) = \frac{\sum_f W^{\mathrm{aqmh}}_{f,c}(p)\, I_{f,c}(p)}{\sum_f W^{\mathrm{aqmh}}_{f,c}(p)}, \quad W^{\mathrm{aqmh}}_{f,c}(p) \geq 0. \tag{1}$$

After the weights have been computed, the reconstruction core is therefore a deterministic linear weighted mean. AQMH does not hallucinate pixel values, does not synthesize replacement frames, and does not rely on tile clustering. Its purpose is narrower: retain useful local regions from every registered frame while suppressing locally degraded regions through a dense, reproducible quality field.

2 Method overview

2.1 Inputs and independence

AQMH reuses only shared preprocessing outputs:

- calibrated, registered, prewarped, and normalized frames;
- a canvas-valid/common-overlap mask;
- global frame diagnostics used for a frame-level weight;
- run management, artifacts, reports, and UI status.

AQMH is otherwise independent from Classic Tile Compile. In particular, the run studied here reports `classic_tile_weights_used=false` and `fallback_to_classic=false`. The classic phases `STATE_CLUSTERING` and `SYNTHETIC_FRAMES` are not part of AQMH reconstruction.

The distinction matters operationally. Shared preprocessing produces registered, normalized image samples and a valid canvas domain. AQMH then defines its own quality maps and reconstruction weights. Any classic tile grid that appears in reports is therefore a geometry or diagnostic artifact, not the AQMH weight lattice.

2.2 Binding invariants

The AQMH implementation follows eight practical invariants:

1. No frame is discarded solely because of its AQMH quality map.
2. Canvas-invalid pixels are excluded from all map and reconstruction statistics.
3. Once weights are known, reconstruction is a linear weighted mean of finite samples.
4. Quality maps and diagnostics are deterministic for identical input data and parameters.
5. AQMH outputs weights, masks, and diagnostics only; it does not predict missing signal.
6. Cherry-pick and any per-pixel selection mode must enforce a documented minimum retained-sample count, below which the mode is disabled automatically.
7. Cross-infrastructure extensions (registration-weight guard, background penalty) may only multiply the global frame weight by a non-negative scalar. They must not alter Q^{map}, must not introduce pixel-dependent weights, and must be independently disableable.
8. Any post-reconstruction neutralisation or detail blending must be validation-gated. The run must record which candidate was selected and why. The final output must not be worse, by the documented regression metrics, than the ordinary AQMH output.

The M31 run uses `cherry_pick_enabled=false`; therefore no optional pixel-level frame subset mode is active.

2.3 Quality-map model

For a normalized registered input frame $I_{f,c}(x, y)$, AQMH computes a multi-scale quality field. The default pyramid uses downscale factors $D_s = 4^s$ with a local radius of four pixels in scale space. Scales that are too coarse for the image dimensions are omitted and the geometric-mean exponent is computed from the actually available scale count. At each retained scale s, three local signals are evaluated over valid canvas support:

- a sharpness signal based on local Laplacian-response variance;
- a local SNR signal based on background-centered positive signal over robust scale;
- an artifact anomaly gate based on high-pass outlier fraction.

The sharpness and SNR signals are normalized within a frame by robust z-scores and combined by a sigmoid:

$$\Psi_s(x, y) = \sigma(w_{\text{sharp}}\, z(\Phi_{\text{sharp},s}) + w_{\text{snr}}\, z(\Phi_{\text{snr},s})) \times \Phi_{\text{artifact},s}(x, y). \tag{2}$$

Scale maps are upsampled with valid-support interpolation and fused by geometric mean:

$$Q^{\text{map}}_{f,c}(x, y) = \left(\prod_{s \in S_{\text{actual}}} \Psi^{\text{up}}_s(x, y) \right)^{1/|S_{\text{actual}}|}. \tag{3}$$

The geometric mean is intentionally conservative: one poor scale suppresses the fused quality at that location, and a zero at any scale remains an exact veto after fusion. Canvas-invalid pixels are set to zero only as an output convention after they have already been excluded from all statistics.

The z-score normalization also fixes the interpretation of Q^{map}. The sigmoid component is a within-frame relative quality field; it indicates which regions of a frame are locally better or worse than other regions of the same frame. Whole-frame discrimination is carried by the global term $G_{f,c}$ and by absolute gates such as the artifact fraction.

2.4 Global quality with background penalty

The effective AQMH weight multiplies the frame-level quality term and the spatial quality map. The v0.2.0 global quality formula introduces an optional background-gradient penalty as a third signal:

$$\begin{aligned} \text{score}_f &= w_{\text{sharp}}\, z_{s,f} + w_{\text{snr}}\, z_{n,f} - w_{\text{bg}}\, z_{b,f} \\ G_{f,c} &= g_{\text{floor}} + (1 - g_{\text{floor}})\, \sigma(\text{clamp}(k_{\text{scale}} \cdot \text{score}_f,\ -8,\ 8)) \end{aligned} \tag{4}$$

where $z_{b,f}$ is the robust z-score of the per-frame sky-gradient summary $g_{\text{background},f} = (q_{\max} - q_{\min})/\text{background}_f$, computed from quadrant-based masked median backgrounds.

A strong large-scale gradient means that the background model used by the SNR signal is a biased local estimate. Frames with very flat backgrounds tend to be cleaner and more linear; frames with a strong moon-glow or light-pollution gradient receive a lower global weight, reducing their influence on the final stack without removing them entirely.

The weights are renormalised so that $w_{\text{sharp}} + w_{\text{snr}} + w_{\text{bg}} = 1$ when at least one effective weight is positive. If a configured weight is ≤ 0, if fewer than three finite positive summary values are available, or if the coefficient of variation $\text{CV} = \text{MAD}/\text{median}$ is below 0.01, the corresponding effective weight is set to zero. This prevents near-constant signals from amplifying noise.

2.5 Registration-weight guard

Before reconstruction, the runner may optionally apply a registration-weight guard to $G_{f,c}$. It reads the shared `global_registration.json` artifact and damps the global weight of frames with low cross-correlation confidence, sequential or predicted registration sources, or deep chain-depth:

$$G^{\text{eff}}_{f,c} = G_{f,c} \cdot R_f, \quad R_f \in [r_{\text{floor}},\ 1]. \tag{5}$$

The guard is a shared-infrastructure multiplier, not part of the AQMH quality model. It prevents high AQMH quality scores from amplifying registration errors.

2.6 Reconstruction

For each output pixel, the implementation accumulates finite, canvas-valid samples only. Missing quality maps are not replaced by unweighted mean contributions. A zero quality map is an explicit veto for that frame/pixel contribution. Iterative sigma clipping with σ_{low} and σ_{high} rejects outlier samples before the weighted mean is computed. A minimum effective sample count $n_{\text{eff}}^{\text{min}}$ and minimum fraction $f_{\min}$ are enforced.

The studied M31 run stores one luminance quality map per frame at half canvas resolution (`resolution_divisor=2`) and interpolates it for reconstruction. This is a storage and I/O choice, not a change to the mathematical definition of AQMH as a dense pixel-weighted reconstruction method.

3 Post-reconstruction robustness extensions

3.1 Uniform control mean

When `compute_uniform_control` is enabled, the reconstruction routine also computes the unweighted mean of the registered frames over the same valid-pixel set:

$$U(p) = \frac{1}{|V_c^I(p)|} \sum_{f \in V_c^I(p)} I_{f,c}(p). \qquad (6)$$

$U(p)$ is not a quality-weighted result; it is a diagnostic reference for the validation gates.

3.2 Validation metrics

Both the raw AQMH output $A(p)$ and any post-processed candidate are compared to $U(p)$ using:

- FWHM regression: $\Delta_{\text{FWHM}} = (\text{FWHM}_{\text{cand}} - \text{FWHM}_{\text{ctrl}})/\text{FWHM}_{\text{ctrl}}$
- Background RMS regression: $\Delta_{\text{RMS}} = (\text{RMS}_{\text{cand}} - \text{RMS}_{\text{ctrl}})/\text{RMS}_{\text{ctrl}}$
- Seam score regression, tail_{11} regression, elongation regression.

Each metric records an applicability state: `pass`, `fail`, or `not_applicable`. A metric is `not_applicable` when either side of the comparison is missing, non-finite, has too few samples, or would require a relative regression against a zero or near-zero control denominator. A `not_applicable` metric is diagnostic only and must not by itself trigger fallback.

3.3 Adaptive low-frequency neutralisation

If $U(p)$ is available, an optional neutralised candidate $N(p)$ is computed:

$$\begin{aligned} L(p) &= \text{GaussianBlur}(A(p) - U(p), \sigma = 96\,\text{px}) \\ N(p) &= A(p) - L(p). \end{aligned} \qquad (7)$$

The blur uses reflect-101 border mode. The sigma of 96.000 px targets large-scale “veil” residuals without affecting structure on smaller scales. $N(p)$ may be selected as the base over $A(p)$ only when the background RMS regression is strictly smaller.

3.4 Structure-masked detail blending

After selecting the base $B(p)$ (either $A(p)$ or $N(p)$), a second optional candidate preserves AQMH detail in high-structure regions while keeping the smoother background from $B(p)$:

$$D(p) = U(p) + M_s(p) \cdot (B(p) - U(p)) \qquad (8)$$

where $M_s(p)$ is a soft structure mask derived from the gradient magnitude of $U(p)$, mapped from the $q_{\text{low}} = 0.40$ quantile to $q_{\text{high}} = 0.90$ quantile to $[0, 1]$ and Gaussian-blurred with $\sigma = 4.000\,\text{px}$. In high-structure regions ($M_s \approx 1$) the candidate follows the AQMH base; in smooth regions ($M_s \approx 0$) it follows the uniform control.

If $D(p)$ fails the applicable validation thresholds, an alpha-blended attenuation search is performed over $\alpha \in [0, 1]$. The largest α that satisfies the thresholds is selected. If no $\alpha > 0$ passes, the base $B(p)$ is kept.

3.5 Final protection gate

After all post-processing, the selected output $O(p)$ is validated one more time against $U(p)$ and the immutable raw AQMH baseline $A(p)$. If any applicable configured threshold is exceeded, the runner rejects the post-processed candidate and emits $A(p)$. Blending toward $U(p)$ is forbidden because it can erase faint diffuse signal while one-sided noise metrics appear to improve.

4 Implementation run

4.1 Data and configuration

The evaluation data are taken from a tile_compile AQMH run on an M31 dataset:

`runs/M31-default-afix1_20260722_111345`

The paper bundle itself is archived under `docs/AQMH/zenodo-0.2.0` in the project repository. Run artifacts used for the measurements are local analysis outputs located in:

`runs/M31-default-afix1_20260722_111345/artifacts`

The final preview shown in Figure 4 is derived from `outputs/stacked_rgb_hms.fits`.

This paper treats the run as an implementation validation case. It verifies that the expected AQMH execution path is followed and reports the resulting diagnostics. It does not compare AQMH against a separately tuned Classic Tile Compile run on the same data, and it does not claim that the reported percentage changes generalize beyond this run.

4.2 Quality-map diagnostics

The run wrote one luma quality map per frame. The average map mean was 0.498 with standard deviation 0.002. The per-frame map mean ranged from 0.492 to 0.504. The average p10 and p90 across frames were 0.377 and 0.636 respectively. The average artifact fraction was 1.920×10^{-6}, with a range from 0.000 to 1.200×10^{-5}. All 645 diagnostics reported the `scene_dependent_snr` flag, which marks use of the degraded local-SNR path when the background-centered support condition is not fully satisfied.

Table 1: M31 run geometry and frame counts.

Quantity	Value
Input frames loaded	645
Usable frames	645
Canvas size	3924 × 2310 px
Common-overlap fraction	0.975
Reconstruction fraction	0.982
Adaptive tiles in shared grid	209
Tile size / stride	296 px / 208 px

Table 2: AQMH map storage and reconstruction support.

Quantity	Value
AQMH enabled	true
Frames/maps	645 / 645
Stored map size	1962 × 1155 px
Resolution divisor	2
Map dtype	uint16
Finite samples	5.350×10^9
Missing samples	0
Unsupported output pixels	18 539.000
Zero-veto pixels	0
Classic fallback	false
Execution backend	cuda_native_v0_2

The global sharpness source was `psf_wfwhm_inverted` for all frames, meaning the PSF-FWHM sharpness proxy was used instead of the Laplacian-variance signal. This avoids a known inversion of `local_variance(laplacian)` on extended-emission targets such as M31, where nebular texture can correlate positively with seeing FWHM. The average global quality was 0.470, ranging from 0.031 to 0.820. The background-penalty input averaged 25.636, reflecting moderate sky-gradient variation across the frame sequence.

4.3 Registration-weight guard

The registration-weight guard was enabled and active. Of the 645 frames, 638 were registered via

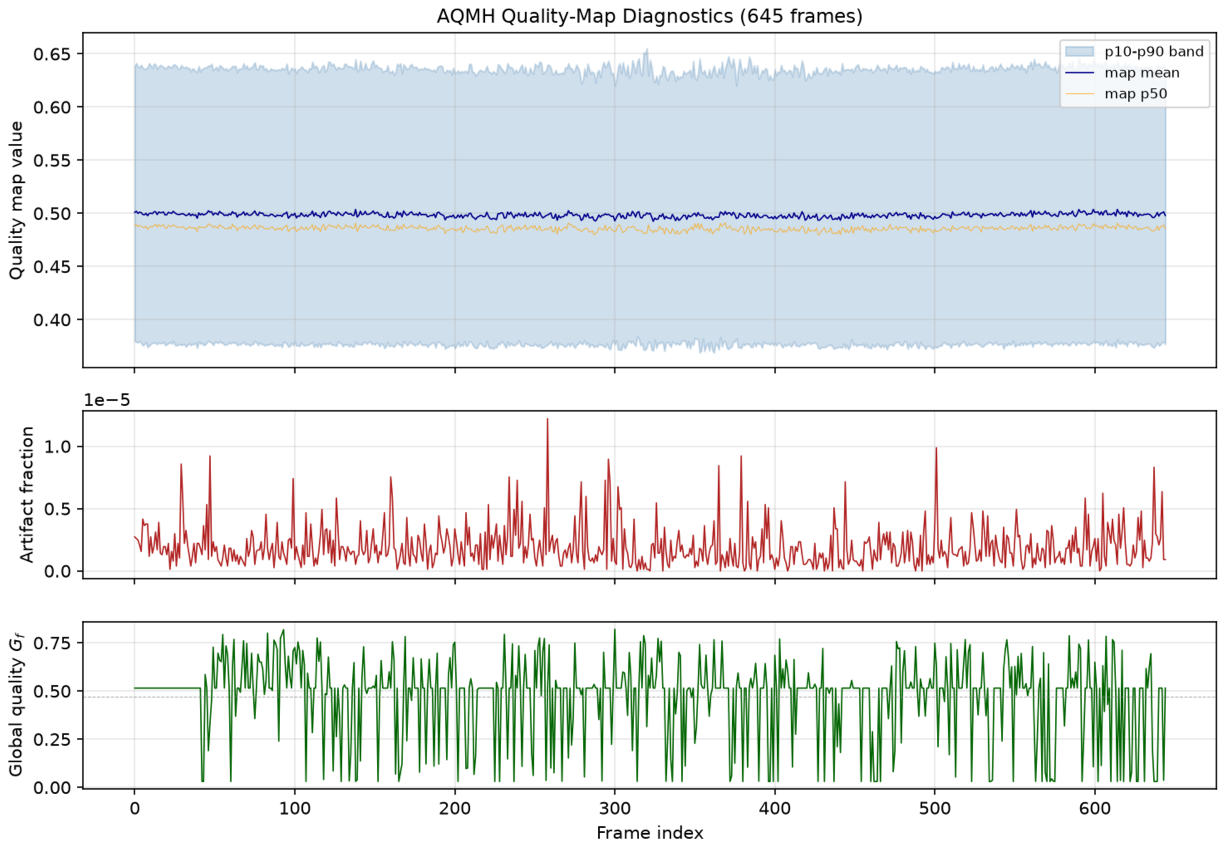

Figure 1: Per-frame AQMH quality-map diagnostics. The shaded band shows the p10-p90 interval of the map values; mean and median remain stable across the 645-frame sequence. The artifact fraction stays near zero, with a higher early-frame tail.

`direct_global`, 6 via `sequential_refined`, and 1 was the reference frame. The guard damped 6 frames (the sequential and reference frames), with a mean factor of 0.998 and a minimum factor of 0.726. The median factor was 1.000, confirming that the vast majority of frames were unaffected.

4.4 Validation metrics

The validation artifact reports an AQMH method run. The median input seeing estimate was 12.329 px, while the median output FWHM was 5.268 px. This corresponds to a 57.273% reduction of the measured FWHM:

$$100\left(1-\frac{5.268}{12.329}\right)=57.273.$$

Background RMS changed from 4.185 to 0.780, an 81.351% decrease.

The uniform-control gate compared the raw AQMH output against the unweighted mean. The results are summarised in Table 4. The FWHM, seam score, elongation, and tail_{11} regressions all passed their thresholds. The background-RMS regression (0.168) exceeded the configured threshold of 0.050, triggering the gate. As a re-

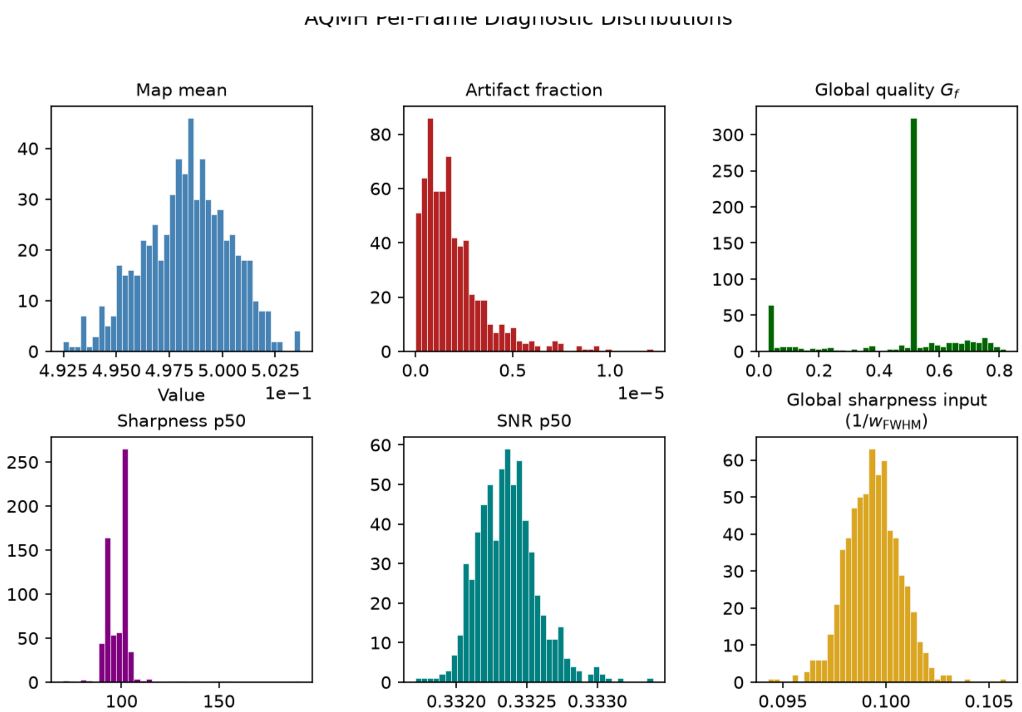

Figure 2: Distributions of AQMH per-frame diagnostic quantities. The narrow quality-map mean distribution reflects the within-frame relative normalization of the sigmoid factor; the artifact gate remains an absolute veto term.

sult, the raw AQMH output was preserved by the guard (`raw_aqmh_preserved_by_guard = true`, `selected_candidate = raw_aqmh`).

The low-frequency neutralisation candidate was evaluated but not applied: its background-RMS regression (0.168) was not strictly smaller than the raw AQMH regression. The structure-masked detail candidate was also evaluated but not applied: while its background-RMS regression (0.016) was smaller, it did not pass all applicable validation thresholds simultaneously. The alpha-blended attenuation search found no $\alpha > 0$ that satisfied all gates, so the base (raw AQMH) was retained.

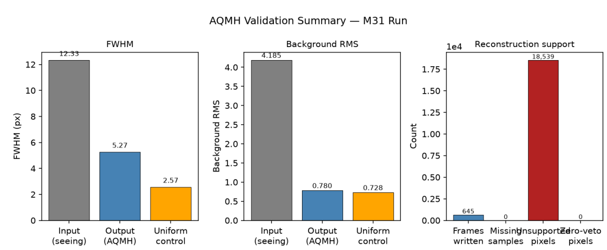

Figure 3: Validation summary for the M31 AQMH run. FWHM improves while the background-RMS regression exceeds the gate threshold, causing the uniform-control gate to trigger. The raw AQMH output is preserved.

Table 3: v0.2.0 extension configuration.

Parameter	Value
Global quality	
g_{floor}	0.03
w_{sharp}	0.55
w_{snr}	0.30
w_{bg}	0.25
k_{scale}	1.5
Registration-weight guard	
Enabled	true
r_{floor}	0.30
$\mathrm{cc}_{\mathrm{floor}}$	0.35
$\mathrm{cc}_{\mathrm{full}}$	0.80
Sequential factor	0.92
Predicted factor	0.50
Reconstruction	
Clip iterations	4
σ_{low}, σ_{high}	2.0, 2.0
f_{min}	0.40
$n_{\mathrm{eff}}^{\mathrm{min}}$	2.0
Neutralisation	
Blur σ	96.000 px
Structure mask	
q_{low}, q_{high}	0.40, 0.90
Blur σ	4.000 px

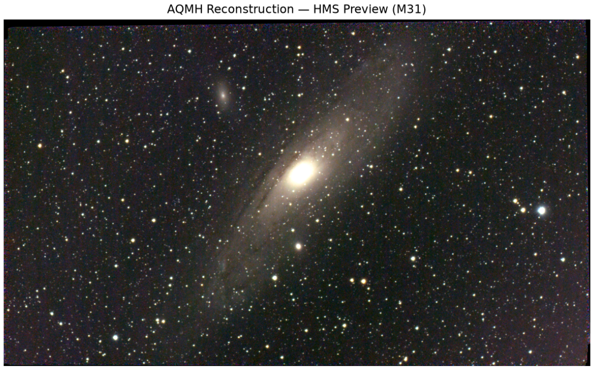

Figure 4: Final RGB preview after AQMH reconstruction, PCC, and HMS. The preview is a visual run product derived from the FITS output, not a quantitative measurement.

Table 4: Uniform-control gate validation results.

Metric	AQMH	Control	Regression
FWHM (px)	2.600	2.573	+0.010
Background RMS	0.850	0.728	+0.168
Seam score	1.736	1.830	−0.051
Elongation	1.098	1.100	−0.001
Tail_{11} abs	0.492	0.494	−0.006

5 Discussion

5.1 What the run demonstrates

The M31 run demonstrates the intended AQMH v0.2.0 execution path with all post-reconstruction extensions active:

- every usable frame produced a quality map;
- the PSF-FWHM sharpness proxy was used for global quality, avoiding the Laplacian-variance inversion on extended-emission targets;
- the registration-weight guard damped weakly-registered frames without affecting the majority;
- reconstruction consumed AQMH maps without classic tile-weight fallback;
- no quality map samples were missing during reconstruction;
- the uniform-control gate correctly identified the background-RMS regression and preserved the raw AQMH output;
- low-frequency neutralisation and structure-masked detail blending were evaluated but rejected by the validation gates, confirming that the guard mechanism works as designed;
- the measured FWHM and background RMS validation metrics improved.

The support counters are particularly important for implementation validation. A successful AQMH-labeled run would be ambiguous if reconstruction silently fell back to missing-map behavior, unweighted means, or classic tile weights. The combination of 645 written maps, zero missing map samples, and `fallback_to_classic=false` makes the measured output attributable to the AQMH path rather than to a mixed or compatibility path.

5.2 Interpretation of the background-RMS gate

The uniform-control gate triggered on the background-RMS regression of 0.168, well above the configured threshold of 0.050. This means the AQMH-weighted output has a 16.800% higher robust background RMS than the unweighted mean. This is not necessarily a defect: the AQMH weighting preserves high-frequency structure that the unweighted mean smooths out, which can increase the pixel-difference-based RMS estimator. The gate's purpose is to detect cases where this increase is excessive; the threshold of 0.050 is conservative. The raw AQMH output was preserved because no post-processed candidate improved all applicable metrics simultaneously.

5.3 Interpretation of stable map means

The frame-to-frame mean of Q^{map} is deliberately stable. The sharpness and SNR terms are z-normalized inside each frame, so their sigmoid contribution is mostly a within-frame relative field. Cross-frame quality discrimination comes from the global frame weight $G_{f,c}$ and from absolute veto terms such as the artifact gate. Therefore, a narrow distribution of map means is not a defect; it is a consequence of separating within-frame spatial quality from between-frame global quality.

5.4 Acceleration

The AQMH_MAPS phase ran on the CPU with 8 parallel workers, while the AQMH_RECONSTRUCTION phase used a native CUDA backend on an NVIDIA GeForce GTX 1660 Ti (6.000 GB VRAM). The reconstruction processed 23.000 chunks of 101.000 rows each, with a device memory budget of 2.330 GB. The total data volume was approximately 52.800 GB, requiring 4.000 sigma-clip iterations per chunk. The effective throughput was 26.600 MB/s, indicating a memory-bandwidth-limited pipeline on this GPU class.

5.5 Limitations

This paper reports one M31 run and should not be read as a full statistical validation across object classes, sensors, exposure lengths, sky brightness levels, and seeing regimes. The diagnostic flag `scene_dependent_snr=true` appears for all frames and deserves further evaluation: the fallback is deterministic, but future work should quantify when background-centered local SNR support is sufficient without the degraded-path flag. The background-RMS gate triggered, which means the raw AQMH output has measurably higher pixel-difference RMS than the uniform control; while this is expected for a quality-weighted reconstruction that preserves high-frequency detail, the threshold calibration deserves sensitivity analysis. The validation metrics are pipeline diagnostics rather than an externally calibrated astronomical quality assessment. The final preview is not a scientific photometry product; it includes downstream PCC and HMS processing.

Further validation should include repeated runs, independent datasets, explicit comparisons against Classic Tile Compile and conventional stackers, sensitivity tests for pyramid parameters and map resolution, and visual/quantitative checks around strong gradients, saturated stars, nebular structure, and low-overlap frame borders.

6 Conclusion

AQMH v0.2.0 provides a deterministic pixel-wise quality-weighted alternative to coarse local tile weighting for short-exposure deep-sky reconstruction. The M31 run confirms that the implementation can execute the complete AQMH path on a 645-frame dataset, write all dense quality maps, apply the registration-weight guard, evaluate the post-reconstruction extension suite (neutralisation, structure-masked blending), and produce improved run-level validation metrics. The observed FWHM reduction of 57.273%, the background RMS reduction of 81.351%, the absence of missing map samples, and the correct operation of the uniform-control gate support the practical viability of AQMH as an independent reconstruction method in tile_compile.

The post-reconstruction extensions (background penalty, registration-weight guard, low-frequency neutralisation, structure-masked detail blending) were all active and correctly evaluated. The validation gates rejected both the neutralised and the structure-masked candidates, preserving the raw AQMH output. This demonstrates that the guard mechanism works as designed: AQMH will not silently replace its output with a post-processed variant that fails the documented regression thresholds.

Code and Data Availability

Reference method. Classic Tile Compile is described by *Tile-Based Quality Reconstruction for DSO,* Version 3.3.9. The corresponding repository source is the v3.3.9 paper source.

Previous version. The AQMH v0.1.0 paper bundle is archived in docs/AQMH/zenodo-0.1.0.

Methodology. The AQMH v0.2.1 methodology document is aqmh_methodik_en_v0.2.1.md.

Code. The public project repository is github.com/jeamy/tile_compile.

This paper. The AQMH v0.2.0 paper bundle is archived in docs/AQMH/zenodo-0.2.0. The compact values used for the tables and plots are stored in m31_aqmh_run_summary.json.

Run data. The full source measurement artifacts are local outputs from an M31 analysis run and are not part of the public repository bundle.