I dug through the notebook and pulled together the exact math, constants, and logic used to compute the “probability of saturation” plot.

Overview

For each redshift z and airmass X (per band), the code estimates P(saturation) = P(central-pixel electrons ≥ full-well) via Monte Carlo over seeing and subpixel centering.

Inputs: SN Ia peak AB magnitude m\_peak(z) from SALT2; seeing FWHM scaling with airmass; AB zeropoint ZP to convert magnitudes to counts; Gaussian PSF integration over the central pixel.

Core Equations

Seeing vs airmass: θ(X) = FWHM0 × X^α with α = 0.6 (geometric FWHM in arcsec).

Ref: Motivation/Rubin\_SN\_Ia\_m5\_SNR\_Saturation.ipynb:163, 164

5σ depth (SMTN-002 fallback):

m5 = C\_m + 0.5 (m\_sky − 21) + 2.5 log10(0.7/θ) + 1.25 log10(t\_vis/30) − k\_atm (X−1)

Ref (markdown): Motivation/Rubin\_SN\_Ia\_m5\_SNR\_Saturation.ipynb:329

Ref (code): Motivation/Rubin\_SN\_Ia\_m5\_SNR\_Saturation.ipynb:560

SNR (fallback when rubin\_sim not available):

SNR ≈ 5 × 10^(−0.4 (m\_peak − m5))

Ref: Motivation/Rubin\_SN\_Ia\_m5\_SNR\_Saturation.ipynb:721, 723

Magnitude to total counts:

counts = 10^((ZP − mag)/2.5)

ZP via rubin\_sim predicted\_zeropoint\_e2v(band, airmass=X, exptime) if available; else fallback ZP\_30s + 2.5 log10(exptime/30)

Ref: Motivation/Rubin\_SN\_Ia\_m5\_SNR\_Saturation.ipynb:628, 633, 636, 639, 640

Gaussian PSF central-pixel fraction:

σ = θ/(2√(2 ln 2)), pixel width p (arcsec), centroid offset (dx, dy) in [−p/2, p/2]

1D fraction along x: f\_x = 0.5 [erf((p/2−dx)/(√2 σ)) − erf((−p/2−dx)/(√2 σ))]; similarly f\_y; 2D fraction f = f\_x f\_y

Ref: Motivation/Rubin\_SN\_Ia\_m5\_SNR\_Saturation.ipynb:646, 649, 651, 652, 653

Central pixel electrons:

e\_central = counts × gain × f

Ref: Motivation/Rubin\_SN\_Ia\_m5\_SNR\_Saturation.ipynb:664, 666, 668

Probability of saturation (Monte Carlo at fixed z, X, band):

P\_sat(z, X) ≈ (1/N) Σ\_{k=1..N} 1{ e\_central(m\_peak(z), X, FWHM\_k, dx\_k, dy\_k) ≥ FW }

FWHM\_k ~ LogNormal(ln θ(X), σ\_lognormal), with σ\_lognormal = 0.15; (dx\_k, dy\_k) ~ Uniform([−p/2, p/2]^2)

Ref: Motivation/Rubin\_SN\_Ia\_m5\_SNR\_Saturation.ipynb:833, 834, 836, 838, 840, 847–859

Assumed Constants

Airmass grid: X ∈ [1.0, 2.0] (50 samples) Motivation/Rubin\_SN\_Ia\_m5\_SNR\_Saturation.ipynb:162

Exposure: EXPTIME = 30 s Motivation/Rubin\_SN\_Ia\_m5\_SNR\_Saturation.ipynb:159

Seeing normalization: FWHM0\_GEOM = 0.7″ at X=1 Motivation/Rubin\_SN\_Ia\_m5\_SNR\_Saturation.ipynb:163

Seeing airmass exponent: α = 0.6 Motivation/Rubin\_SN\_Ia\_m5\_SNR\_Saturation.ipynb:164

Pixel scale: p = 0.2″/pixel Motivation/Rubin\_SN\_Ia\_m5\_SNR\_Saturation.ipynb:174

Gain: GAIN\_E\_PER\_ADU = 1.0 Motivation/Rubin\_SN\_Ia\_m5\_SNR\_Saturation.ipynb:177

Full-well “hardcap”: FW = 80,000 e− (warn at 64,000 e−) Motivation/Rubin\_SN\_Ia\_m5\_SNR\_Saturation.ipynb:175, 176

Subpixel mode for probability: random-single (one random subpixel per trial) Motivation/Rubin\_SN\_Ia\_m5\_SNR\_Saturation.ipynb:853

MC trials: N\_MC = 1000 Motivation/Rubin\_SN\_Ia\_m5\_SNR\_Saturation.ipynb:833

Seeing scatter: SEEING\_SIGMA\_FRAC = 0.15 (lognormal RMS ~15%) Motivation/Rubin\_SN\_Ia\_m5\_SNR\_Saturation.ipynb:834

SMTN-002 coefficients and sky (fallback m5):

C\_m: u=22.97, g=24.58, r=24.60, i=24.54, z=24.37, y=23.84 Motivation/Rubin\_SN\_Ia\_m5\_SNR\_Saturation.ipynb:553

k\_atm: u=0.47, g=0.21, r=0.13, i=0.10, z=0.07, y=0.17 Motivation/Rubin\_SN\_Ia\_m5\_SNR\_Saturation.ipynb:554

Dark sky mag/arcsec²: u=22.0, g=22.4, r=21.2, i=20.5, z=19.6, y=18.6 Motivation/Rubin\_SN\_Ia\_m5\_SNR\_Saturation.ipynb:555

Fallback AB zeropoints (30 s): u=25.0, g=26.3, r=26.4, i=26.1, z=25.4, y=24.7 Motivation/Rubin\_SN\_Ia\_m5\_SNR\_Saturation.ipynb:626

SALT2 peak setup: x1=0, c=0, MB=−19.3 (Vega B) and compute min AB magnitude per band over phase Motivation/Rubin\_SN\_Ia\_m5\_SNR\_Saturation.ipynb:284, 299, 301, 306, 307, 308

Logic Flow

Load Rubin total bandpasses and register with SNCosmo; compute m\_peak(z) per band by scanning phases and taking the minimum magnitude. Motivation/Rubin\_SN\_Ia\_m5\_SNR\_Saturation.ipynb:284

Compute m5(X) via rubin\_sim if available; else use SMTN-002 formula with θ(X)=FWHM0×X^0.6. Motivation/Rubin\_SN\_Ia\_m5\_SNR\_Saturation.ipynb:566, 584, 592, 595, 597

For probability heatmap at band b:

For each grid point (z, X): look up m\_peak(z). Motivation/Rubin\_SN\_Ia\_m5\_SNR\_Saturation.ipynb:846

Repeat N\_MC times: draw FWHM ~ LogNormal(ln θ(X), 0.15), draw a random subpixel offset (dx, dy), convert m\_peak to counts via ZP, compute e\_central using Gaussian pixel integral, compare to FW, and count hits. Motivation/Rubin\_SN\_Ia\_m5\_SNR\_Saturation.ipynb:847–859

Probability P = hits/N\_MC; assemble into P(X, z) and plot as an image. Motivation/Rubin\_SN\_Ia\_m5\_SNR\_Saturation.ipynb:871–879

Notes

rubin\_sim improves realism by providing bandpasses, m5 via calc\_m5, SNR via calc\_snr\_m5, and predicted zeropoints; otherwise the code uses documented analytic fallbacks.

Probability depends on the assumed seeing scatter and subpixel centering; PSF is Gaussian (not Moffat), and only the central pixel’s full-well is checked.

SNR is not used to compute P(saturation); it’s reported alongside for context.

If you want, I can extract and display a compact, formula-only sheet derived from the notebook defaults for quick reference or adjust any of the assumptions (e.g., centered pixel, different full-well, or a different seeing scatter) and regenerate the logic.