What Bloats Your Floats? Right-Sizing Numerical Precision for Scientific Computing

Pratyai Mazumder
ETH Zurich
Zurich, Switzerland
pmazumder@ethz.ch

Alexandru Calotoiu
ETH Zurich
Zurich, Switzerland
acalotoiu@ethz.ch

Marcin Copik
ETH Zurich
Zurich, Switzerland
mcopik@ethz.ch

Yakup Koray Budanaz
ETH Zurich
Zurich, Switzerland
ybudanaz@ethz.ch

Shriram Chandran
ETH Zurich
Zurich, Switzerland
schandran@ethz.ch

Torsten Hoefler
ETH Zurich
Zurich, Switzerland
htor@ethz.ch

***Abstract*—While FP64 (*binary64*) remains the standard representation of real numbers in scientific computing, hardware trends increasingly favor low-precision formats optimized for AI workloads. This shift prioritizes throughput and energy efficiency over full-precision capabilities. We challenge the necessity of high-precision arithmetic in PDE-governed simulations, where spatial and temporal discretization errors introduce a physical noise floor that masks the lower-order bits of the FP64 format, rendering them irrelevant. We propose a criterion based on the Signal-to-Noise Ratio (SNR) with respect to spatial and temporal refinements to evaluate the impact of safe precision reduction on scientific computations. Finally, we present an automated dataflow-centric workflow for targeted lowering within mixed-precision subgraphs in complex scientific applications. Large-scale evaluations of weather and climate applications show that strategic precision reduction on GPUs achieves up to $1.8\times$ speedup without compromising physical fidelity.**

***Index Terms*—Floating-point arithmetic, numerical stability, approximation error, partial differential equations, signal to noise ratio, data flow computing, atmospheric modeling, high performance computing**

I. Introduction

The FP64 format has long been the default for large-scale scientific applications [1], [2], historically deemed necessary for computational stability [3], [4]. Modern hardware, however, is moving in the opposite direction: driven by AI workloads, GPUs increasingly optimize for FP32, FP16, BF16, and FP8, relegating native FP64 to a marginal fraction of peak throughput [5], [6]. For memory-bandwidth-bound PDE solvers—the computational core of weather and climate models—remaining locked to 64-bit data structures exacerbates the memory wall and bottlenecks throughput [7], [8], [9]. Because data movement dominates execution time [9], [10], halving the bit-width effectively doubles memory bandwidth [11], [12], providing a direct path to higher-resolution or larger-ensemble simulations [11], [12], [13], [14].

From a physical standpoint, the case for FP64 is weaker than commonly assumed. At practical grid resolutions, spatial and temporal discretization errors dominate the solution error budget, masking floating-point round-off and rendering the lower-order significand bits of FP64 irrelevant [12], [13], [15], [16], [17], [18]. Yet production codes remain conservatively locked to FP64 [19], [20], [21], [22], [23]—not because the physics demands it, but because the community lacks scalable tools to validate that lower precision is safe [24].

The validation bottleneck is twofold. First, large-scale simulation frameworks consist of millions of lines of legacy code with complex numerical dependencies; naively downcasting variables across regions with heterogeneous error sensitivities can introduce instability, catastrophic cancellation, or solver divergence [25], [26]. The combinatorial explosion of assigning precisions to thousands of variables renders manual optimization intractable [2], [27], and existing auto-tuning methods fail to scale to monolithic codebases [24].

Second, existing precision-tuning frameworks rely on the wrong evaluation methodology. They bound pointwise errors against a high-precision reference trajectory [19], [28], [29], but in PDE solvers this reference shares the same discretization error as the test run. The resulting metrics measure the arithmetic gap between formats—not the physical accuracy of either solution. In chaotic systems, acceptable grid-scale noise causes these pointwise metrics to saturate rapidly [30], [31], [32], rendering tuning constraints overly pessimistic and sometimes misleading [12], [33], [34].

Compounding the tooling gap is a measurement gap. Validating precision reductions requires collapsing multidimensional error fields into a robust scalar metric, yet the standard choices are ill-suited to chaotic PDE solvers. Pointwise L_p norms grow exponentially under chaotic trajectory divergence [30], [31], [32]; relative errors hit zero-crossing singularities [35]; and distributional metrics such as Wasserstein distance or CRPS [11], [12], [14], while physically meaningful, are too expensive to embed in an automated tuning loop. What is needed is a metric that is scale-free, tolerant of nonlinear dynamics, and computable from a single execution—without requiring a reference beyond FP64.

© 2026 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.

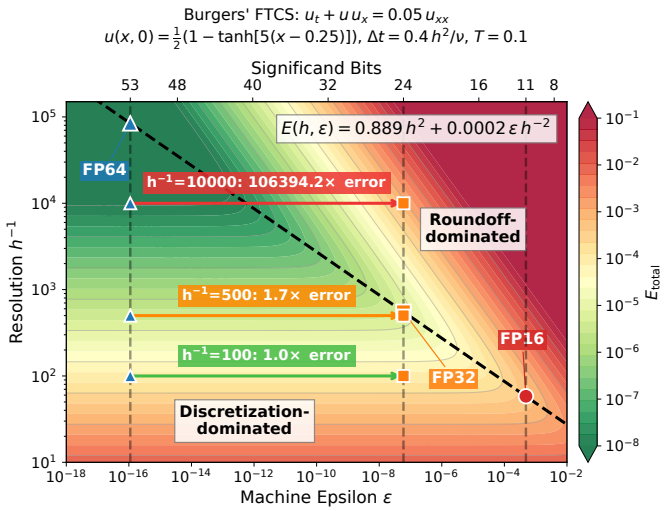


Fig. 1. Total error landscape $E_{\text{total}} = \Theta(h^2) + \Theta(\epsilon\, h^{-2})$ for forward-time centered-space (FTCS) scheme on the 1D viscous Burgers' equation ($\nu = 0.05$, $T = 0.1$), demonstrating that the impact of lowering precision depends strongly on the operating grid resolution h^{-1}. The dashed line marks the optimal resolution locus $h_{\text{opt}}^{-1}(\epsilon)$. Arrows indicate the error penalty when switching from FP64 to FP32 at three distinct grid resolutions: the error increase is negligible at a coarse resolution of $h^{-1} = 100$ (green), grows to 1.7× at $h^{-1} = 500$ (orange), and suffers a catastrophic $\sim 10^5\times$ increase at a fine resolution of $h^{-1} = 10\,000$ (red).

A. Contributions

C1. A New Perspective on Precision Tuning in Scientific Computing: We propose a variance-based Signal-to-Noise Ratio (SNR) criterion that replaces the conventional high-precision reference with the discretization noise floor as the evaluation baseline. This threshold is derived from spatial and temporal refinements alone, reflects the intrinsic accuracy limit of the discretization, and satisfies the requirements above: it is scale-free, tolerant of nonlinear dynamics, and needs no reference beyond FP64.

C2. Automated Dataflow-Level Precision Tuning in Production-Scale Weather and Climate Models: We build an automated framework that performs systematic precision reduction directly on the dataflow graph and validates safety against the proposed threshold. We demonstrate its efficacy on the ECMWF CLOUDSC microphysics scheme [36] and the ICON velocity tendency kernel [31], achieving up to 1.8× GPU speedup while preserving macroscopic numerical fidelity.

II. Discretization Noise and Precision Limits

In scientific computing, physical phenomena are modeled by PDEs, and solved using discretization-based techniques like finite difference, finite volume, and finite element methods. While our work is motivated by weather and climate codes, the proposed techniques apply broadly across numerical solvers.

The total error of a PDE solver is conventionally bounded by the sum of the norms of the modeling error E_{model}, the discretization error E_{disc}, and the round-off error E_{ro}:

$$\|u^* - \hat{u}_{h,\tau}\| \leq \|E_{\text{model}}\| + \|E_{\text{disc}}\| + \|E_{\text{ro}}\|, \tag{1}$$

where u^* is the true physical state and $\hat{u}_{h,\tau}$ is the computed solution using floating-point arithmetic for a uniform grid resolution h and timestep τ.

For a given physical model, we ignore E_{model} and are concerned with $E_{\text{total}} \equiv E_{\text{disc}} + E_{\text{ro}}$. As a result, the computational accuracy is governed by the balance between the spatial discretization error and the round-off error. Consider a PDE of order m, with a numerical scheme of spatial order p and temporal order q. The spatial and temporal discretization errors are $\mathcal{O}(h^p)$ and $\mathcal{O}(\tau^q)$, respectively. Because deterministic round-off error accumulates at a rate proportional to the number of timesteps, the round-off error is $\mathcal{O}(\epsilon\tau^{-1})$ at every grid point, provided that $\tau^{-1}\epsilon \ll 1$ [3, Ch. 3]. Using continuous-equivalent discrete norms, one can show this same asymptotic bound extends to the global round-off error [37, App. A]. Consequently,

$$\|E_{\text{disc}}\| + \|E_{\text{ro}}\| \leq \mathcal{O}(h^p) + \mathcal{O}(\tau^q) + \mathcal{O}\left(\frac{\epsilon}{\tau}\right). \tag{2}$$

We make two standard assumptions: (1) the stability constraints hold $\tau = \mathcal{O}(h^m)$, and (2) the scheme is balanced such that spatial error asymptotically bounds or matches temporal error ($p \leq qm$) [37]. Then, minimizing the total error yields the optimal configuration: $h_{\text{opt}} = \mathcal{O}(\epsilon^{1/(p+m)})$.

However, hardware constraints restrict extreme-scale simulations to a much coarser resolution ($h_{\text{prac}} \gg h_{\text{opt}}$) [38]. For instance, reaching h_{opt} for FP64 in hydrodynamic simulations requires problem sizes orders of magnitude beyond exascale capabilities (e.g., 286 billion grid cells for a 1D Sod's shock tube) [16]. We can define the precision $\epsilon^* = \mathcal{O}(h_{\text{prac}}^{p+m})$ for a given resolution h_{prac} at which the round-off error matches the discretization error. Clearly, at a precision $\epsilon \leq \epsilon^*$, we can control the total error asymptotically with only the discretization error: $\|E_{\text{disc}}\| \leq \|E_{\text{total}}\| \leq 2\|E_{\text{disc}}\|$.

Figure 1 illustrates this balance between h and ϵ for the 1D viscous Burgers' equation. At relatively coarse resolutions (e.g., $h^{-1} \leq 500$), the penalty for reducing precision from FP64 to FP32 is almost negligible. Only at extreme refinement (e.g., $h^{-1} = 10\,000$), well past the optimal locus, does the penalty become catastrophic.

A. The Marginal Utility of Precision

To understand the utility of precision beyond ϵ^*, we model the round-off error as *noise*. Although FP arithmetic is deterministic, the propagation of round-off errors in fluctuating dynamic signals is classically modeled via Pseudo-Quantization Noise (PQN) theory [43], [44]. Under this PQN framework, the round-off error $E_{\text{ro}}(\epsilon)$ for a given precision ϵ behaves as a zero-mean stochastic variable that is uncorrelated with the discretization error E_{disc}.

The statistics of the total error E_{total} reduce to a mean of E_{disc} and a variance of $\mathbb{E}[\|E_{\text{ro}}(\epsilon)\|^2]$, and the second moment accumulates in quadrature:

$$\mathbb{E}[\|E_{\text{total}}(\epsilon)\|^2] = \|E_{\text{disc}}\|^2 + \mathbb{E}[\|E_{\text{ro}}(\epsilon)\|^2]. \tag{3}$$

Furthermore, the round-off variance scales quadratically with machine epsilon: $\mathbb{E}[\|E_{\text{ro}}(\epsilon)\|^2] \propto \epsilon^2$ [43, Ch. 12].

TABLE I

Comparison of scalar, field-level error metrics for evaluating reduced-precision arithmetic in nonlinear PDEs. Computational cost is $\mathcal{O}(N)$ in all cases except the distributional metrics, which incur higher computational cost. ✓ = yes, ∼ = partial, ✗ = no. Columns: **Scale-free** = Dimensionless; adapts to dynamic range without manual normalization or zero-crossing pathologies; **Nonlinear-safe** = Tolerates shocks and chaotic divergence without false saturation; **Mean-invariant** = Insensitive to static mean shifts; evaluates anomalies rather than absolute states.

Metric	Scale-free	Non-linear-safe	Mean-invariant
Pointwise & norm-based			
Max relative error ($\|e/u\|_{L_\infty}$; ULPs)	∼ [39]	✗[1]	✗
Absolute max error ($\|e\|_{L_\infty}$)	✗	✗[1]	✗
Global error norms ($\|e\|_{L_1}$, $\|e\|_{L_2}$; RMSE)	✗	∼[1]	✗
Distributional			
Wasserstein (optimal transport)	✗	✓	✗ [40]
CRPS (ensemble scoring)	✗	✓	✗
Hellinger (f-divergence)	✓	✓	✗
KS distance (supremum CDF gap)	✓	✓	✗
Perceptual & Structural			
SSIM (image quality)	✗ [41]	∼	✗ [42]
DSSIM (data SSIM)	✓ [41]	∼	✗ [42]
Signal-to-noise			
$\mathrm{SNR}_{\mathrm{pow}}$	✓	∼	✗
$\mathrm{SNR}_{\mathrm{var}}$	✓	✓	✓

[1] Pointwise metrics fail to converge at shocks $\mathcal{O}(1)$ [37], and rapidly saturate globally under chaotic trajectory divergence [13], [32]. (L_1 provides partial tolerance by integrating $|e|$ rather than e^2).

Next, we evaluate the utility of any higher precision format $\epsilon_k = 2^{-k}\epsilon^*$ using k additional significand bits. Recalling that $\mathbb{E}[\|E_{\mathrm{ro}}(\epsilon^*)\|^2] = \|E_{\mathrm{disc}}\|^2$ as per the definition of ϵ^*, the second moment of the total error becomes:

$$\mathbb{E}[\|E_{\mathrm{total}}(\epsilon_k)\|^2] = \|E_{\mathrm{disc}}\|^2 \left(1 + 4^{-k}\right). \tag{4}$$

The marginal variance reduction provided by the k-th additional bit after ϵ^* decays exponentially as $\mathcal{O}(4^{-k})$:

$$\Delta_k = \mathbb{E}[\|E_{\mathrm{total}}(\epsilon_{k-1})\|^2] - \mathbb{E}[\|E_{\mathrm{total}}(\epsilon_k)\|^2] = 3 \cdot 4^{-k}\|E_{\mathrm{disc}}\|^2. \tag{5}$$

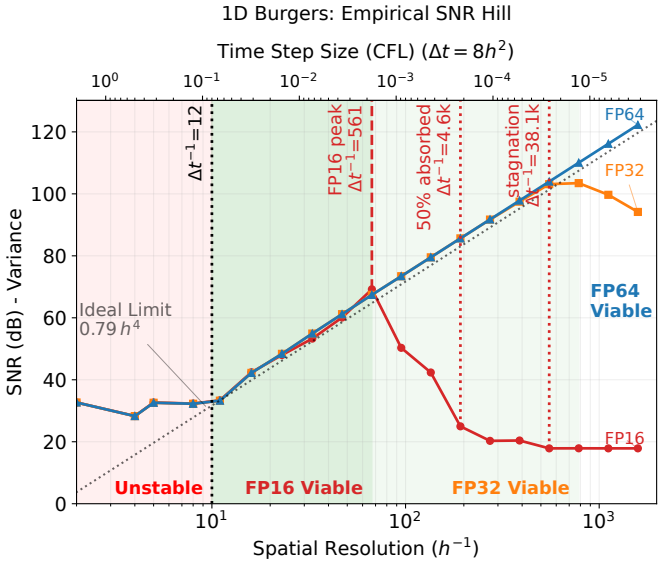
Consequently, the total error converges exponentially to the discretization floor $\|E_{\mathrm{disc}}\|$. The additional significand bits required to reach ϵ_{64} from ϵ^* provide diminishing returns for the macroscopic solution fidelity.

B. Physical Limits and Error Quantification

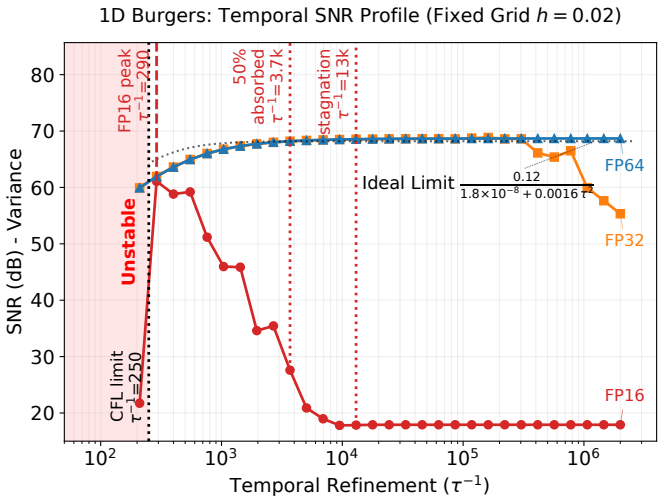
To quantify numerical fidelity, the multidimensional error field must be collapsed into a scalar metric. As summarized in Table I, traditional deterministic bounds fail to capture the physical stability of nonlinear PDEs. By treating round-off error E_{ro} as *noise*, we transition to the Signal-to-Noise Ratio (SNR). The conventional power-based SNR is:

$$\mathrm{SNR}_{\mathrm{pow}}[u^*, \hat{u}_{h,\tau}] = \frac{\mathbb{E}[\|u^*\|^2]}{\mathbb{E}[\|u^* - \hat{u}_{h,\tau}\|^2]}, \tag{6}$$

where u^* and $\hat{u}_{h,\tau}$ are the analytical and computed solutions, respectively. Under spatial averaging, $\mathrm{SNR}_{\mathrm{pow}}$ corresponds strictly to the inverse squared relative L_2 norm. While this expectation-based formulation effectively reframes geometric



(a) Spatial refinement at a proportional CFL ratio.



(b) Temporal refinement at a fixed spatial resolution ($h = 0.02$).

Fig. 2. SNR curves for the 1D viscous Burgers' equation across three precisions. Lines show empirical SNR; the dotted *Ideal Limit* is the precision-independent discretization floor. (a) All precisions track the ideal $\mathcal{O}(h^{-4})$ slope until round-off ($\propto \varepsilon^2 h^{-4}$) dominates, producing a peaked hill that shifts rightward with precision. Past the peak, SNR does not collapse immediately: numerical viscosity partially absorbs round-off energy, creating a plateau between the "50% absorbed" and "stagnation" markers before full degradation sets in. (b) At fixed h, temporal refinement improves SNR as $\mathcal{O}(\tau^{-2})$ until saturating at the spatial discretization floor, then degrades through temporal round-off ($\propto \varepsilon^2 \tau^{-2}$), with an analogous absorption plateau for FP32.

distance as stochastic noise power, both $\mathrm{SNR}_{\mathrm{pow}}$ and the L_2 norm remain highly sensitive to harmless static biases.

Because dynamic systems are driven by spatiotemporal gradients rather than absolute magnitudes, the variance-based SNR [45] discards the bias to promote the structural preservation of the dynamic anomalies and fluctuations that govern the simulated physics:

$$\mathrm{SNR}_{\mathrm{var}}[u^*, \hat{u}_{h,\tau}] = \frac{\mathrm{Var}[u^*]}{\mathrm{Var}[u^* - \hat{u}_{h,\tau}]}. \tag{7}$$

For uncorrelated discretization and round-off errors, the denominator of Equation (7) is decomposed as:

$$\mathrm{Var}[E_{\mathrm{total}}] \equiv \mathrm{Var}[u^* - \hat{u}_{h,\tau}] = \mathrm{Var}[E_{\mathrm{disc}}(h)] + \mathrm{Var}[E_{\mathrm{ro}}(\epsilon)]. \tag{8}$$

For a spatial discretization of order p, $\mathrm{Var}[E_{\mathrm{disc}}(h)] = C_1 h^{2p}$. Under the PQN framework, the round-off variance scales as $\mathrm{Var}[E_{\mathrm{ro}}(\epsilon)] = C_2 \epsilon^2 h^{-2m}$. Substituting these into Equation (7) yields the scaling relationship:

$$\mathrm{SNR}_{\mathrm{var}}(\epsilon, h) = \frac{\mathrm{Var}[u^*]}{C_1 h^{2p} + C_2 \epsilon^2 h^{-2m}}, \tag{9}$$

where C_1 and C_2 are problem-dependent proportionality constants, and often unknowable analytically. Asymptotically, Equation (9) suggests the "hill" shape of SNR-vs.-h plot in Figure 2a, and maximizing the function yields an optimal h that is the maximum refinement a given precision ϵ can support. In the limit of infinite precision, the round-off noise vanishes, and we recover a more standard scaling curve:

$$\lim_{\epsilon \to 0} \mathrm{SNR}_{\mathrm{var}}(\epsilon, h) = \frac{\mathrm{Var}[u^*]}{C_1 h^{2p}} = \Theta(h^{-2p}).$$

Even in a purely chaotic system subject only to perturbation error, decorrelated trajectories saturate at twice the natural variance [46], [47], bounding $\mathrm{SNR}_{\mathrm{var}}$ within $[-3, \infty)$ dB. Thus, $\mathrm{SNR}_{\mathrm{var}}$ serves as an effective descriptor of the system's inherent "noise" in a scale-free manner.

C. An Example: SNRs in 1D Burgers' Equation

To validate the theoretical bounds, we consider the 1D viscous Burgers' equation, $u_t + u\,u_x = \nu\,u_{xx}$, a canonical nonlinear parabolic PDE ($d = 1$, $m = 2$) that couples advective steepening with viscous diffusion [48]. We discretize this PDE using a forward-time centered-space (FTCS) scheme ($p = 2$, $q = 1$), whose diffusive stability limit constrains $\tau = \mathcal{O}(h^2)$. This scheme is deliberately minimal: its central-difference advection term is unconditionally unstable and relies entirely on physical viscosity for damping [37, Ch. 10], making it acutely sensitive to grid-scale noise injected by finite-precision arithmetic.

Figure 2 presents the resulting SNR curves. In Figure 2a, the empirical SNR profiles (SNR $\propto E^{-2}$) all closely track the $\mathcal{O}(h^{-4})$ spatial discretization limit at practical resolutions ($h^{-1} < 50$), confirming that surplus precision offers no measurable benefit to solution fidelity. An SNR collapse only occurs during extreme over-refinement, where accumulated round-off error ($\propto \varepsilon^2 h^{-4}$) overtakes the discretization limit. The same holds in time (Figure 2b): refinement improves SNR only until the spatial floor caps it.

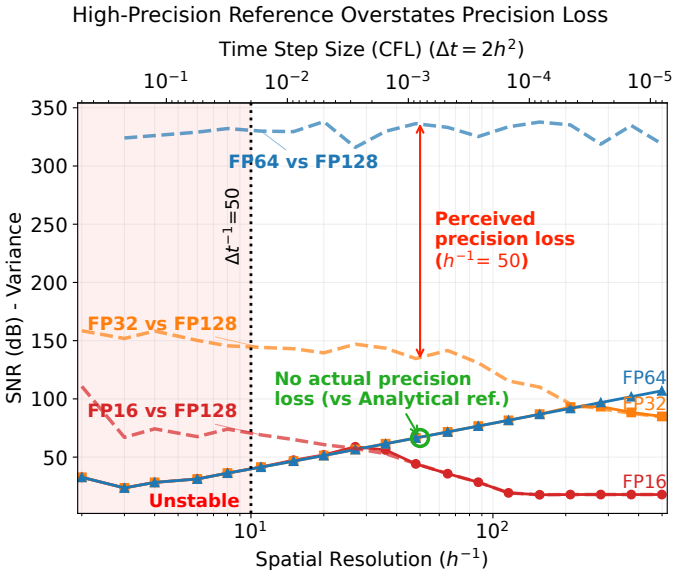


Fig. 3. SNR of the 1-D Burgers FTCS solution measured against two baselines: the analytical solution (solid) and an FP128 numerical reference on the same grid (dashed). The red arrow marks the apparent FP32-to-FP64 gap in the reference curves at $h^{-1} = 50$; the green circle marks the corresponding analytical SNR, where FP64 and FP32 are indistinguishable.

D. Moving Beyond Arithmetic Baselines

Standard floating-point tuning frameworks compare mixed-precision outputs against a high-precision reference (e.g., FP128) on the same grid [19], [25]. Under this methodology, switching from FP64 to FP32 reveals a sharp SNR drop—on the order of 180 dB—that appears catastrophic. But the drop is an artifact: because the reference shares the same grid, it inherits the same E_{disc}, so the SNR measures only the arithmetic gap between formats, not the physical accuracy of either [12], [13]. Even with infinite precision the solver cannot exceed the discretization ceiling; extra significand bits merely reproduce truncation artifacts more faithfully.

Figure 3 confirms this. At practical resolutions ($h^{-1} \leq 50$), the FP128-referenced curves (dashed) show a large FP64-to-FP32 gap, yet the analytical ones (solid) put both at the same $\sim$68 dB floor. Only FP16 falls below it, a genuine precision limitation. Section II-F estimates this floor without u^*.

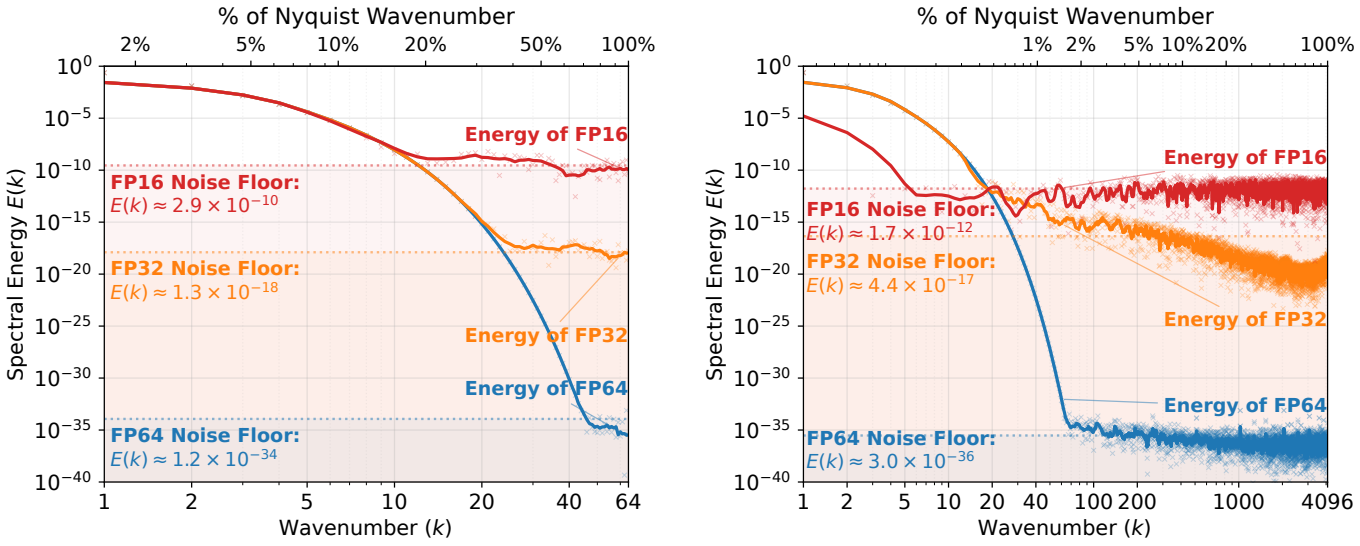
E. Spectral Analysis and Stability

Noise and signal are best separated in the frequency domain. By Parseval's theorem, the spatial variance $\mathrm{Var}[u]$ equals the integral of the power spectral density $E(k)$ up to the Nyquist limit $k_{\max} = \pi/h$ [43, Ch. 12], [37, Ch. 9]:

$$\mathrm{Var}[u] = \int_0^{k_{\max}} E(k)\,dk. \tag{10}$$

Round-off error, modeled as white noise, has a flat power spectral density E_{noise}. The accumulated variance $\mathrm{Var}[E_{\mathrm{ro}}] \propto \epsilon^2 h^{-2m}$ is distributed over the d-dimensional spectral volume $V_k \propto h^{-d}$, so the noise floor scales as $E_{\mathrm{noise}}(\epsilon, h) \propto \epsilon^2 h^{d-2m}$.

Let $E_{\mathrm{phys}}(k)$ be the power spectral density of the exact solution u^*. Physical signals conventionally exhibit power-law spectral decay in multiscale dynamical systems [49],



(a) Coarse grid ($N_x = 128$)

(b) Fine grid ($N_x = 8192$)

Fig. 4. Spectral energy $E(k)$ of the velocity field (1D viscous Burgers', FTCS) at three precisions. Dotted lines and shading mark each precision's noise floor ($\propto \epsilon^2/N$). (a) For $N_x = 128$, FP16 meets its noise floor near $\sim$30% of Nyquist and flattens, while FP32 and FP64 remain signal-dominated. (b) For $N_x = 8192$, FP16 saturates near $\sim$1% of Nyquist, triggering a "tail curl" that feeds artificial energy back into macroscopic scales.

[50], [51]. For such signals characterized by a spectral decay rate $\alpha > 0$ such that $E_{\text{phys}}(k) \propto k^{-\alpha}$, the intersection with the noise floor defines a critical wavenumber $k_{\text{crit}} \propto (\epsilon^2 h^{d-2m})^{-1/\alpha}$. Beyond k_{crit}, the spectrum saturates at the spectral noise level, representing an unphysical energy accumulation at the grid scale. However, empirical evidence demonstrates that this elevated noise floor remains isolated at the truncation limit without interacting with or prematurely degrading the macroscopic scales [11].

Figure 4 visualizes this threshold in the frequency domain. The physical energy cascade $E(k)$ decays at high wavenumbers, while floating-point round-off imposes a flat spectral noise floor $\propto \epsilon^2/N_x$ [43, Ch. 12, 20]. Where that floor meets the cascade decides whether the noise is harmless. At coarse resolution it lands where viscous damping has already removed the energy. At fine resolution it intersects the cascade before dissipation sets in, and nonlinear interactions carry the grid-scale excess back into the resolved physical modes [11], [17], [31]. This confirms that precision can be safely reduced whenever the solver operates in the discretization-limited regime.

F. Empirical Estimation via Refinement

Because the exact analytical solution u^* is unavailable for complex PDEs [37, Ch. 1], the true physical noise floor (SNR_{true}) cannot be computed directly. Instead, we estimate this limit empirically via full grid refinement, and show that temporal-only refinement provides a strict, practical overestimate that bypasses some challenges of spatial refinement.

Given spatial and temporal orders of accuracy p and q, the true discretization error E_{disc} in the asymptotic regime expands as the sum of leading-order truncation terms:

$$E_{\text{disc}}(h,\tau) = \tilde{u}_{h,\tau} - u^* = C_x h^p + C_t \tau^q + \mathcal{O}(h^{p+1}, \tau^{q+1}), \tag{11}$$

where C_x and C_t are problem-dependent coefficients. By evaluating the solver at the target discretization (h,τ) and comparing the resulting state with a proportionally refined run $(h/r_x, \tau/r_t)$, we isolate these leading-order terms in a computable difference field [37, App. A]:

$$\begin{aligned}\Delta\tilde{u}_{h,\tau} &= \tilde{u}_{h,\tau} - \tilde{u}_{h/r_x,\tau/r_t} \\ &\approx C_x h^p \left(1 - \frac{1}{r_x^p}\right) + C_t \tau^q \left(1 - \frac{1}{r_t^q}\right).\end{aligned} \tag{12}$$

Because the refinement factors satisfy $r_x, r_t > 1$, the variance of this computable difference field underestimates the true discretization error variance ($\text{Var}(\Delta\tilde{u}_{h,\tau}) < \text{Var}(E_{\text{disc}})$). This proxy thus yields a strictly conservative, upper-bound estimate of the grid's noise floor:

$$\text{SNR}_{\text{grid}} = \frac{\text{Var}(\tilde{u}_{h,\tau})}{\text{Var}(\Delta\tilde{u}_{h,\tau})} > \frac{\text{Var}(\tilde{u}_{h,\tau})}{\text{Var}(E_{\text{disc}})} = \text{SNR}_{\text{true}}. \tag{13}$$

While SNR_{grid} provides a sound threshold, full spatial refinement in large, complex models (e.g., 3D Earth systems) is often impossible when relying on empirical real-world data. Conversely, restricting refinement entirely to the temporal domain ($r_t > 1$, $r_x = 1$) is trivial in time-marching schemes:

$$\Delta\tilde{u}_{\tau} = \tilde{u}_{h,\tau} - \tilde{u}_{h,\tau/r_t} \approx C_t \tau^q \left(1 - \frac{1}{r_t^q}\right). \tag{14}$$

Because stability constraints tie τ to h, $\Delta\tilde{u}_{\tau}$ captures only a fraction of the total discretization error ($\text{Var}(\Delta\tilde{u}_{\tau}) \leq \text{Var}(\Delta\tilde{u}_{h,\tau})$), establishing a stricter noise floor:

$$\text{SNR}_{\text{time}} = \frac{\text{Var}(\tilde{u}_{h,\tau})}{\text{Var}(\Delta\tilde{u}_{t})} \geq \text{SNR}_{\text{grid}} > \text{SNR}_{\text{true}}. \tag{15}$$

Figure 5 validates this ordering on the 1D Burgers' equation, where the analytical u^* makes SNR_{true} computable. As

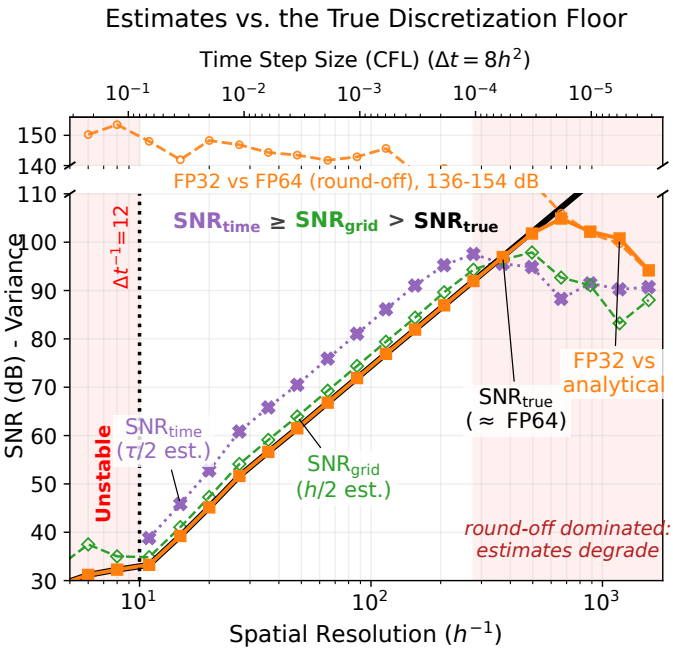


Fig. 5. Floor estimates for the 1D viscous Burgers' equation. The temporal ($\tau/2$) and spatial ($h/2$) estimates bound the true floor from above across the discretization-limited range, confirming $\mathrm{SNR}_{\mathrm{time}} \geq \mathrm{SNR}_{\mathrm{grid}} > \mathrm{SNR}_{\mathrm{true}}$. Both degrade beyond $h^{-1} \approx 300$, where the difference fields are themselves round-off dominated. The top strip is the same-grid FP32-vs-FP64 comparison, off-scale at 136–154 dB: an arithmetic reference registers none of this structure.

Section IV shows, justifying the precision-lowering to FP32 suffices in practice. Since we only need to differentiate broad precision tiers (e.g., FP64 vs. FP32), this trivial temporal over-estimate is a highly practical, black-box criterion.

Therefore, our fundamental criterion for safe precision reduction dictates that noise from precision reduction must not degrade the signal below the full grid noise floor ($\mathrm{SNR}_{\mathrm{grid}}$). In practice, however, this operative threshold is satisfied by enforcing our stricter temporal over-estimate:

$$\mathrm{SNR}_{\mathrm{var}}(\tilde{u}_{h,\tau}, \hat{u}_{h,\tau}) \geq \mathrm{SNR}_{\mathrm{time}}. \tag{16}$$

III. Automated Precision Reduction Framework

While Section II establishes the theoretical viability of precision lowering, manually downcasting complex scientific codebases is labor-intensive and error-prone [2]. Automating this requires a compilation framework that exposes an intermediate representation (IR) separating control and dataflow. We utilize the Data-Centric (DaCe) framework [52], whose dataflow IR provides the required granularity. DaCe represents programs as dataflow graphs, allowing us to target specific data containers for precision reduction—including intermediate transients that are invisible at the application interface. By modifying the types of these containers, dataflow analysis automatically tracks the downstream impact, enabling targeted precision reduction within specific computational subgraphs without invasive source code rewrites [23], [53].

This automated pipeline (Figure 6) begins by extracting the target subcomputation from the surrounding application and translating it into the DaCe IR. It first runs a baseline noise analysis on coarse- and fine-resolution pairs, computing the discretization noise floor ($\mathrm{SNR}_{\mathrm{grid}}$, or $\mathrm{SNR}_{\mathrm{time}}$ where spatial refinement is unavailable) for each output container.

With this objective baseline established, the framework uniformly downcasts the entire dataflow subgraph to the target precision, wherever permitted by the compiler, and executes a verification pass against the serialized reference. We deliberately choose uniform over per-variable mixed precision: the coarse granularity avoids the combinatorial search over individual containers and scales to large codebases, at the cost of optimality (Section VI). If the noise of the lowered subgraph does not degrade the output SNR below the objective baseline, the target precision is validated as safe. Otherwise, the framework reports the failing fields and their SNR deficits, and the user can accept the baseline precision or investigate at finer granularity. The framework then compiles the optimized subgraph and generates the "glue code" for reintegration, so the host application needs no source-code changes.

Offloading to the GPU is another transformation on the same graph (Figure 6), applied alongside the downcast but orthogonal to it. The framework itself runs once, offline: a few baseline and verification runs plus the lowering compilation. Its instrumentation adds a one-time initialization to the deployed run, amortized across timesteps. Beyond the boundary casts of Section III-A, there is no per-timestep cost.

A. Type Semantics and Application Interfacing

Transforming a dataflow graph requires adapting arithmetic operations to modified storage types. Because the DaCe IR maps to C++, the framework leverages standard C++ type promotion rules rather than statically forcing operation types. The downcast targets all floating-point data containers—including intermediate accumulators and reduction temporaries—but leaves integer loop indices and control variables unchanged. Operations with fully narrowed inputs execute natively at the reduced precision (ϵ_{low}). In mixed-precision scenarios involving both base (ϵ_{base}) and lowered (ϵ_{low}) operands, implicit widening to ϵ_{base} preserves intermediate accuracy and mitigates significance loss at internal precision boundaries.

If the entire application is captured within the IR, precision transitions are handled by the compiler. However, when lowering an isolated subgraph that must execute within a surrounding ϵ_{base} application, the framework must carefully manage the data interface. Typically, this requires explicit boundary casts: inputs are narrowed on ingestion ($\hat{x} = \mathrm{cast}(x, \epsilon_{\mathrm{low}})$), outputs widened on return ($y = \mathrm{cast}(\hat{y}, \epsilon_{\mathrm{base}})$). This isolates the subgraph's noise as a bounded perturbation, leaving the host's memory state intact.

Crucially, these boundary casts are not always necessary. Dataflow analysis detects when a lowered container is purely internal to the subgraph (e.g., a transient scratchpad buffer), eliminating the casting overhead. However, when boundary casts are unavoidable, they add per-invocation overhead at the subgraph interface. Each such cast is one streaming pass over a boundary array. These passes are bandwidth-bound,

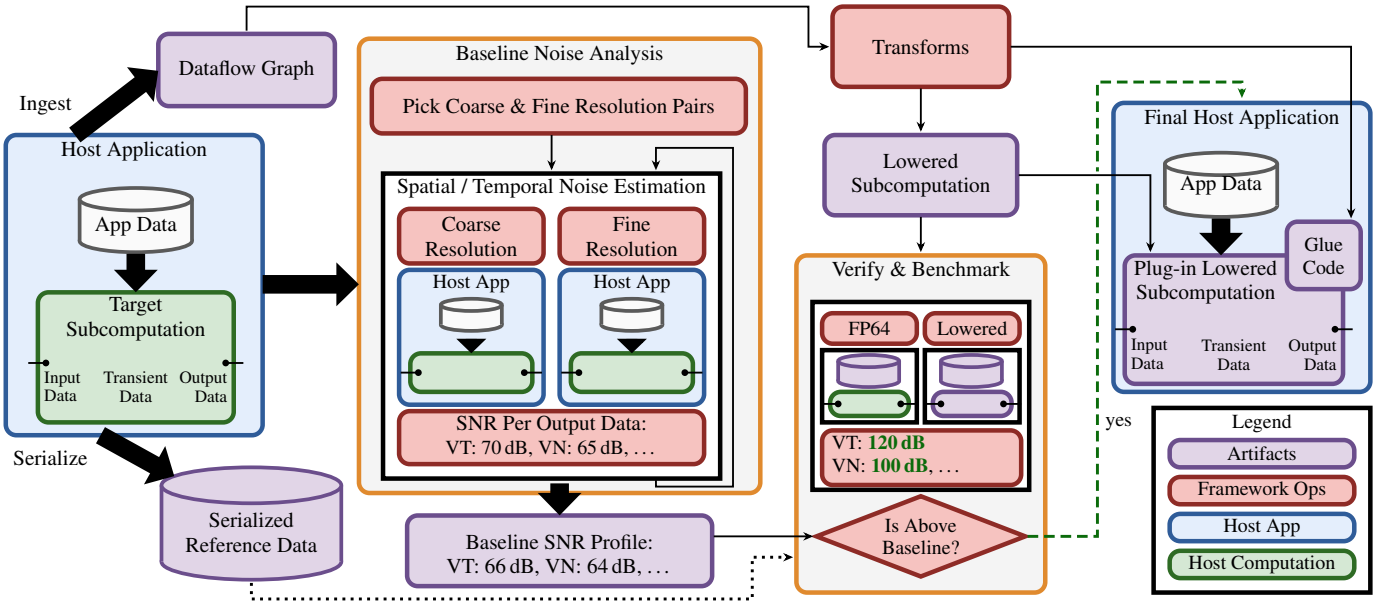


Fig. 6. Automated Precision Lowering Workflow. A target subcomputation is isolated and translated into the DaCe IR alongside serialized reference data. A baseline noise analysis establishes the discretization noise floor using spatial or temporal refinement. The framework then uniformly downcasts the target subgraph to the desired precision and verifies that the resulting output SNR remains strictly above this baseline. The optimized kernel is subsequently wrapped in automatically generated glue code and reintegrated into the surrounding application.

comparable to a copy kernel. Widening the lowered scope—ultimately to the full application—eliminates these transitions entirely. The overhead is therefore an artifact of lowering a region in isolation, not of the method itself. Note that analogous promotion overhead arises when the target hardware lacks native support for specific operations at ϵ_{low} (e.g., FP16 transcendentals); this effect is discussed in Section VI.

IV. Evaluation

This section evaluates automated precision lowering by addressing two questions: (1) Can precision be reduced without compromising physical fidelity? and (2) Does the safe precision tier yield meaningful performance gains? We first establish the discretization noise floors of our target applications and evaluate numerical accuracy across the FP64, FP32, FP16 and BF16 formats. We then quantify the execution performance of the validated configurations, and close by comparing the criterion against the fixed tolerances and sensitivity bounds that existing tools require.

A. Experiment Setup

1) Applications: We evaluate two operational weather model components: Velocity Tendencies (VT) from the ICON dynamical core [31], and the ECMWF CLOUDSC microphysics scheme [36]. These represent the primary computational motifs in Earth-system modeling: memory-bound unstructured grid stencils (VT) and compute-bound column physics (CLOUDSC). VT is targeted specifically because it constitutes a dominant performance bottleneck within ICON.

VT evaluates the advective terms of the momentum equation in ICON's non-hydrostatic dynamical core, written in vector-invariant form: the kinetic-energy gradient, the vorticity flux, and vertical advection. It also computes the wind diagnostics these terms require. On the unstructured mesh, every such operator becomes an indirect stencil. ICON labels that mesh RnBk. Each edge of the base icosahedron is split into n segments, and the resulting triangles are recursively bisected k times, giving $20n^2 4^k$ cells. Every increment of k therefore halves the mean cell width and quadruples the cell count. We evaluate R02B03–R02B06 (320, 160, 80 and 40 km; 5,120 to 327,680 cells) at 90 vertical levels. VT is ingested as an SDFG and processed through the full automated pipeline (Section III), serving as the end-to-end demonstration of framework-driven precision lowering.

CLOUDSC parameterizes cloud microphysics. It advances temperature, specific humidity, cloud fraction and the condensate types—liquid, ice, rain and snow—through the conversions among them: condensation, autoconversion, accretion, freezing, melting and sedimentation. Vertical columns are independent. We evaluate it in its existing single-precision build mode, which validates the SNR methodology on a second, structurally distinct application. The kernel advances one physics timestep ($t_{\text{phys}} = 120\,\text{s}$) by forward Euler integration, split into N equal steps. Performance runs use $N = 10$; the SNR evolution of Figure 7b uses $N = 10{,}000$, integrating the same 120 s at a $1000\times$ smaller step.

2) System and Compilation: Experiments are conducted on single nodes of two CSCS systems: a node of the Alps system equipped with an NVIDIA GH200 Grace Hopper Superchip (96 GB HBM3, 4 TB/s peak bandwidth) [54] and CUDA 12.6; and an Ault node equipped with an NVIDIA A100 SXM4 (40 GB HBM2e, 1.56 TB/s peak bandwidth) and CUDA 12.9.

Applications are compiled into shared libraries from DaCe-generated CUDA code using `nvcc -O3`

TABLE II
Per-field variance SNR [dB] for ICON velocity tendencies precision modes vs. FP64 baseline at four grid resolutions, after p=2 physics timesteps. Underlined entries indicate that the reduced-precision mode falls below $\mathrm{SNR}_{\mathrm{time}}$ at that resolution (i.e., no longer viable).

	vn	w	vt	vn_ie	w_concorr_c
320 km Resolution					
FP32	154.0	139.7	151.2	151.0	151.4
FP16	75.7	61.8	73.0	72.8	82.3
BF16	57.8	44.4	54.9	54.9	64.8
$\mathrm{SNR}_{\mathrm{time}}$	58.8	31.1	65.3	61.5	58.7
160 km Resolution					
FP32	154.1	135.4	151.3	151.1	151.4
FP16	75.8	57.4	73.0	72.9	82.7
BF16	57.8	39.9	55.0	55.0	64.8
$\mathrm{SNR}_{\mathrm{time}}$	66.6	30.4	72.4	69.2	69.9
80 km Resolution					
FP32	154.1	131.6	151.3	151.1	151.4
FP16	75.8	53.6	73.0	73.0	82.8
BF16	57.8	36.2	55.0	55.1	64.5
$\mathrm{SNR}_{\mathrm{time}}$	73.2	30.2	79.0	76.5	77.0
40 km Resolution					
FP32	154.1	126.9	151.3	151.1	151.4
FP16	75.8	48.7	73.0	73.0	82.7
BF16	57.8	31.4	55.0	55.1	64.6
$\mathrm{SNR}_{\mathrm{time}}$	80.9	30.1	86.9	83.8	85.1

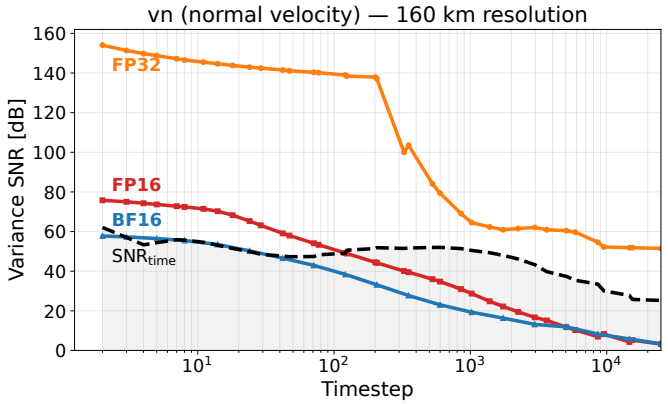
`--use_fast_math`, targeting `sm_90` on GH200 and `sm_80` on A100. Precision lowering, GPU offloading, and boundary cast insertions are managed by the automated DaCe pipeline (Section III). Reproducibility scripts and configuration details are in the Artifact Description appendix.

3) Execution and Methodology: All benchmarking follows established best practices for reproducible performance reporting [55]. Hardware profiling (e.g., bandwidth, utilization) is collected on GH200 via `ncu --set full` in a standalone test harness that executes a single computational step from a serialized input snapshot, isolating the kernel from host framework interference; warmup iterations are excluded from profiling. CLOUDSC wall-clock timings are measured directly within this harness, while ICON-VT timings are extracted from full model runs due to tight coupling with the dynamical core. All timings exclude boundary casting overhead (i.e., FP64-to-FP32/FP16/BF16 interface conversions).

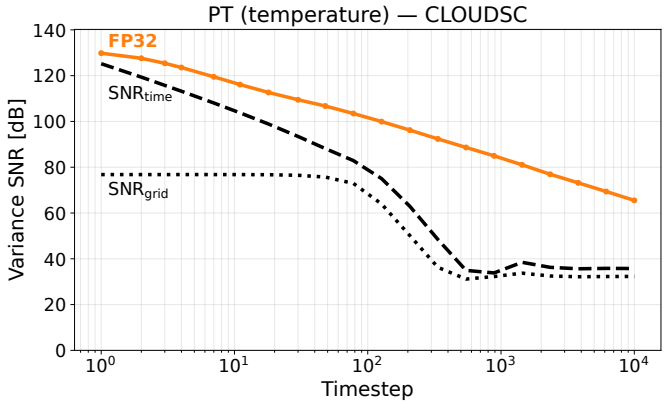
Numerical quality is quantified by the variance SNR relative to the FP64 baseline. A target precision is deemed safe only if its SNR remains above the discretization noise floor (Section IV-B). For ICON-VT, this baseline is estimated via temporal refinement ($\mathrm{SNR}_{\mathrm{time}}$); for CLOUDSC, temporal ($\mathrm{SNR}_{\mathrm{time}}$) and vertical ($\mathrm{SNR}_{\mathrm{grid}}$) refinement baselines are computed.

B. Numerical Accuracy

We estimate each baseline noise level by comparing against refined configurations (Section II-F). While temporal refinement is straightforward for explicit schemes, spatial refinement is challenging for unstructured grids like ICON's. Because $\mathrm{SNR}_{\mathrm{time}}$ proves sufficient to confirm the viability of FP32 (which clearly preserves accuracy on all grids), and of FP16 on coarser grids, we rely on the temporal baseline for ICON-



(a) ICON-VT: vn SNR over 25k physics steps (160 km, $\tau_{\mathrm{phys}} = 8$ s). FP32 ($\sim 50 - 154$ dB) stays above the temporal floor ($\sim 25 - 62$ dB) throughout. Its sharp drop near step $\sim$200 marks the point where roundoff saturates, well above the floor. FP16 ($\sim 3 - 76$ dB) falls below the floor within $\sim$200 steps, and BF16 ($\sim 3 - 58$ dB) starts at it.



(b) CLOUDSC: PT SNR over 10,000 steps for a 120 s interval ($\Delta t = 12$ ms). FP32 ($\sim 65 - 128$ dB) starts just above the temporal floor $\mathrm{SNR}_{\mathrm{time}}$ ($\sim 35 - 125$ dB) and pulls $\sim$30 dB clear as that floor falls. The spatial floor $\mathrm{SNR}_{\mathrm{grid}}$ ($\sim 32 - 77$ dB) lies below both: flat while profiles are smooth, then dropping sharply when a cloud layer forms ($\sim$step $100 - 300$).

Fig. 7. $\mathrm{SNR}_{\mathrm{var}}$ evolution over the full runs. In both applications the floors fall as the step count grows, and FP32 stays clear throughout.

TABLE III
Per-field variance SNR [dB] for CLOUDSC precision modes vs. FP64 baseline at p=2, $t_{\mathrm{phys}} = 120$ s. Values are independent of column count (164k–656k) as CLOUDSC operates column-wise on identical input. Temporal discretization reference from nsub = 1 vs. 2; grid reference from k_{lev} 137 vs. 274 with nsub = 1 vs. 2, refining both discretizations together. "–" indicates divergence to NaN.

Field	FP32	FP16	$\mathrm{SNR}_{\mathrm{time}}$	$\mathrm{SNR}_{\mathrm{grid}}$
PT	124.2	–	112.3	46.4
PQ	116.2	–	92.1	45.1
PA	93.5	3.8	75.8	26.3
PCLV	81.4	-0.2	12.0	10.4

TABLE IV
ncu profiling for ICON velocity tendencies (VT) on GH200, warm rep; cold and warm differ by $<5\%$ on all metrics. Wall-clock times are in Figure 8 — ncu locks clocks to base frequencies, so its timings are not production-comparable.

FP	Memory			Compute share (%)‡		
	Read (MB)	Write (MB)†	BW util. (%)	FP64	FP32	FP16/ BF16
320 km Resolution						
FP64	118	0.1	27.2	95.2	2.9	1.9
FP32	60.0	0.0	15.7	0	91.0	9.0
FP16	31.4	0.0	8.2	0	13.5	86.5
BF16	31.4	0.0	8.2	0	27.3	72.7
160 km Resolution						
FP64	461	45.8	51.8	95.2	2.9	1.9
FP32	233	8.1	31.3	0	92.0	8.0
FP16	121	0.6	15.7	0	15.6	84.4
BF16	121	0.3	15.4	0	28.0	72.0
80 km Resolution						
FP64	1,850	461	70.5	95.2	2.9	1.9
FP32	931	187	47.3	0	92.0	8.0
FP16	483	58.5	23.7	0	15.3	84.7
BF16	483	58.9	22.9	0	27.9	72.1
40 km Resolution						
FP64	11,350	3,850	75.2	95.7	2.0	2.4
FP32	5,220	1,860	51.6	0	93.5	6.5
FP16	2,660	913	28.6	0	17.0	83.0
BF16	2,660	915	27.7	0	28.5	71.5

† DRAM write measures L2 → DRAM writebacks. At 320 km the working set fits in L2 (96 MB on GH200), so stores are absorbed entirely by L2 with no DRAM evictions during the kernel.
‡ FMA (fused multiply-add) counted as 2 FLops [56].

VT. Conversely, CLOUDSC's 1D column physics features a simple, horizontally invariant structure, permitting a straightforward vertical refinement to obtain a meaningful $\mathrm{SNR}_{\mathrm{grid}}$.

Figure 7 tracks the variance SNR over successive timesteps for one prognostic variable from each application—vn for ICON-VT and PT for CLOUDSC—while Table II and Table III give it across all primary output fields. This profiling exposes prognostic fields that are inherently noise-dominated at the grid scale. For example, the vertical velocity (w) in ICON-VT and cloud condensate (PCLV) in CLOUDSC stabilize at floors of $\sim$30 dB and $\sim$12 dB even in FP64. Discretization noise already eclipses the lower-order significand bits for such variables, so a large error there is expected: a lowering is safe only if its output SNR stays above these baselines.

Evaluated against this criterion, FP32 is safe for both applications. In CLOUDSC, PT and PQ retain FP32 SNRs of 124.2 dB and 116.2 dB, comfortably above both their temporal (112.3 dB, 92.1 dB) and vertical grid baselines (46.4 dB, 45.1 dB). All evaluated ICON-VT fields clear their respective $\mathrm{SNR}_{\mathrm{time}}$ thresholds by wide margins in single precision.

FP16 presents a mixed picture. In CLOUDSC, uniform FP16 catastrophically diverges to NaN for PT and PQ, and its SNR for PA drops to 3.8 dB—falling well below both discretization floors. In ICON-VT, a uniform FP16 pass identified w_concorr_c as the sole variable that diverges immediately when lowered; retaining it at FP32 while lowering all remaining containers to FP16 yields the configuration shown in Table II. All fields clear their $\mathrm{SNR}_{\mathrm{time}}$ thresholds at coarse resolutions, but the rising discretization floor overtakes FP16's noise level at 80 km and beyond (underlined entries). These are two-step verdicts. Figure 7a shows that the coarse-grid margin is not durable: at 160 km, FP16 crosses the floor within $\sim$200 steps, so its admissibility here holds for short integrations only.

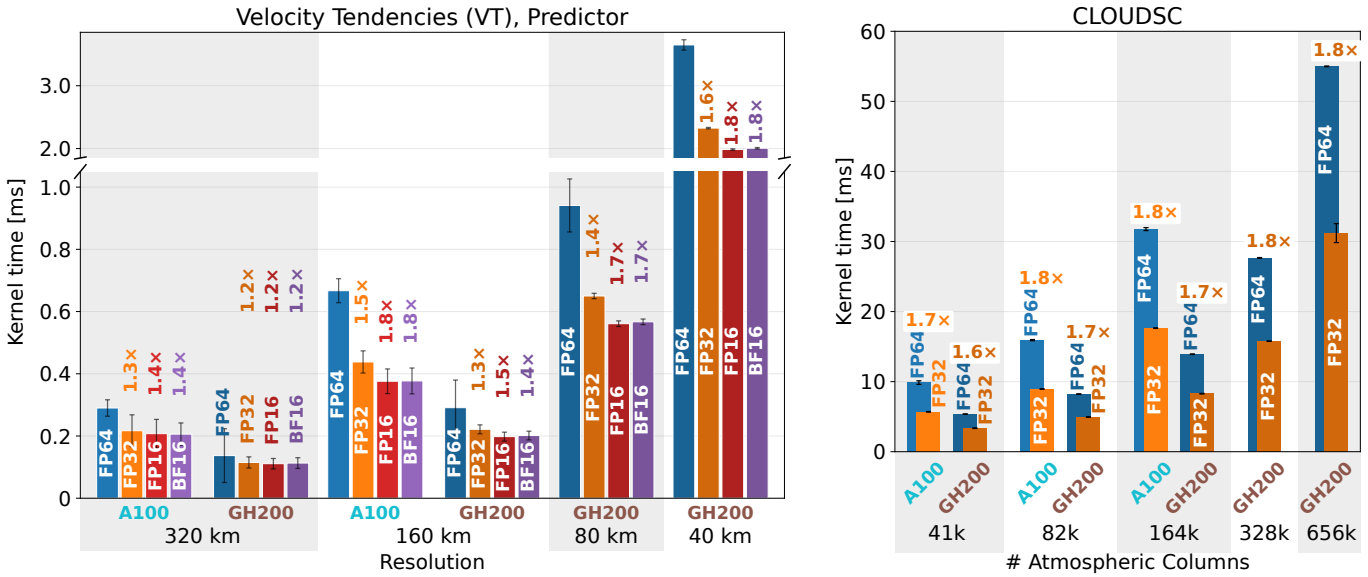
BF16 fares worse still. Across every field and resolution it sits a uniform $\sim$18 dB below FP16—three fewer significand bits at $\sim$6 dB. Only w remains above $\mathrm{SNR}_{\mathrm{time}}$ at all four resolutions, and w_concorr_c only at 320 km.

C. Computational Performance

Having established that FP32 preserves numerical fidelity on both applications, while FP16 is safe only on the coarser VT grids and fails outright in CLOUDSC, we now quantify the resulting performance impact. Figure 8 details execution times and relative speedups for the precision-lowered VT and CLOUDSC kernels. VT evaluates a predictor and a corrector path in each dynamics substep; we report the predictor, as the corrector shows the same speedup profile. Transitioning from FP64 to FP32 yields substantial performance improvements: CLOUDSC gains 1.6–1.8$\times$ on GH200 and 1.7–1.8$\times$ on A100, while VT rises from 1.2$\times$ at 320 km to 1.6$\times$ at 40 km on GH200, and reaches 1.3–1.5$\times$ on A100. Because these kernels are memory-bandwidth-bound, halving the data element size halves DRAM traffic, allowing performance gains to approach the theoretical 2$\times$ limit as the problem size grows.

For FP16, CLOUDSC is ruled out entirely. For VT, FP16 is the fastest configuration measured, reaching 1.8$\times$ at 40 km on GH200 and 1.8$\times$ at 160 km on A100, and BF16 tracks it to within 0.1$\times$ everywhere, since the two move identical memory traffic. The advantage is resolution-dependent: at 320 km both merely match FP32. The largest gains also fall outside the admissible window, as Table II clears FP16 only at 320 and 160 km and BF16 only for w. At 160 km the two coincide, with FP16 both admissible over short integrations and faster than FP32 (1.5$\times$ against 1.3$\times$ on GH200).

Hardware profiling data (Table IV) confirms this plateau. For ICON-VT, downcasting from FP64 to FP32 halves DRAM reads (e.g., 1,850 MB to 931 MB at 80 km) and increases L1/L2 cache hit rates due to the compressed memory footprint. However, the transition from FP32 to FP16 is not matched by a comparable speedup despite DRAM traffic halving further to 483 MB. Profiling reveals that as data volume shrinks, DRAM bandwidth utilization drops from 70.5% at FP64 to 23.7% at FP16 for 80 km, and to 8.2% for 320 km. This indicates that at FP16, the kernel is no longer constrained by main memory bandwidth; consequently, further compressing the memory footprint provides no execution time benefit. The compute mix



(a) ICON velocity tendencies (VT; aquaplanet) at four resolutions. Each dynamics substep evaluates a predictor and a corrector path; we report the predictor, as the corrector's speedups match. BF16 tracks FP16 to within 0.1×, as the two move identical memory traffic.

(b) CLOUDSC microphysics (41k–656k columns). FP16 is omitted due to numerical divergence (Section IV-B).

Fig. 8. Host-side GPU kernel wall-clock time on A100 and GH200; all possible data lowered to the respective precision; speedup relative to FP64. A100 covers only the grids that fit its 40 GB HBM (VT up to 160 km; CLOUDSC up to 164k columns); larger grids are GH200-only.

compounds this: whereas the FP64 build issues 95% of its floating-point work in FP64 and the FP32 build 92% in FP32, the FP16 build retains roughly a seventh at FP32, because the `w_concorr_c` chain (Section IV-B) underflows in half precision and is held there. BF16 moves FP16's memory traffic but retains nearly twice that share at FP32, since division and square root lack native bfloat16 instructions.

D. Field-Dependent Thresholds

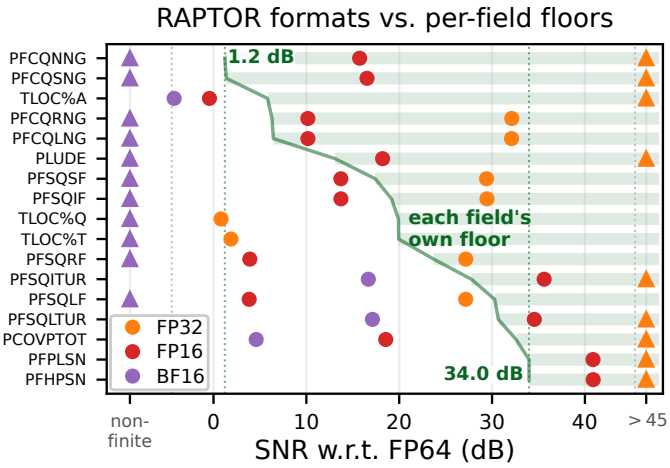
We now contrast our method with two tools representing the state of the art in precision lowering: RAPTOR [21] and ADAPT [57]. The existing tools measure the effect of precision lowering, but leave the fundamental question of what deviation is acceptable open to the user: RAPTOR renders a lowered execution to inspect, ADAPT demands an error bound as input. We demonstrate that the underlying principle of our work addresses this gap of missing tolerance.

RAPTOR: RAPTOR emulates a format by truncating every arithmetic operation. An LLVM instrumentation pass replaces floating-point operations in user-selected scopes (function, file, or program) with calls into a runtime that executes them via MPFR at a user-specified exponent and significand width, casting the result back to the original type after every operation. The tool leaves to the user the decision of what deviation from the full-precision run is tolerable. Furthermore, RAPTOR's truncation is scope-based (function, file, or program), while our dataflow-based approach permits finer-grained control over mixed precision (e.g., per data container).

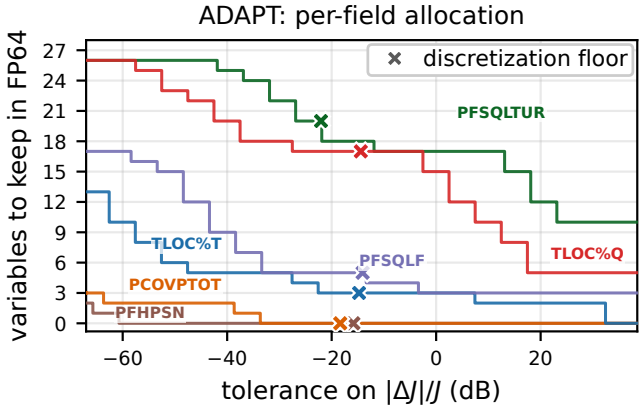
We emulate FP32, FP16 and BF16 with RAPTOR and compare every output against the FP64 run in $\mathrm{SNR}_{\mathrm{var}}$ (Figure 9a). The SNRs improve with more significand bits wherever the lowered run stays finite, but the distribution differs from field to field. RAPTOR, at its best, can use a predefined threshold to declare safety of the lowering (i.e. equivalent to draw a vertical line on Figure 9a). But how much error do we already accept at FP64? That becomes clearer only when we account for the estimated discretization floor, spanning from 1.2 to 34.0 dB across 17 lowered fields. FP32 falls below it for 3 of them, FP16 for 8, and BF16 for all 17. We can conclude that those fields must remain in higher precision when lowering.

ADAPT: ADAPT ranks variables by adjoint sensitivity against a given bound. The user specifies a scalar quantity of interest J (e.g., norm of a field or residual), and a set of candidate variables for demotion from the working precision to a single lower one—FP64 to FP32 in practice. A single run under reverse-mode algorithmic differentiation yields, for every candidate x, the sensitivity $\partial J/\partial x$ at each point where x is assigned. ADAPT combines these sensitivities with the rounding error the lower precision would introduce into x to obtain a first-order estimate of the resulting error in J, and marks variables for demotion in order of increasing estimated impact until a user-specified bound on $|\Delta J|$ is exhausted. The bound is an absolute deviation of J from the FP64 run. This is not compatible with the SNR approach we take, but the underlying principle is demonstrably useful even in this case.

We run ADAPT once per output field, with that field as its sole dependent, and sweep the error bound it is given (Figure 9b). Tightening the bound makes ADAPT keep more of the 55 demotable arrays in FP64, and the curve is different for every field. As before, choosing where on that curve to sit is left to the user, and this is where our simple refinement gives an estimated bound on how much error is tolerable for every



(a) SNR of each output field against the FP64 run: one row per field, one marker per format emulated by RAPTOR. The green step is that field's floor, obtained by evaluating the kernel at 137 and at 274 vertical levels with the substep count held fixed. Triangles at the left edge mark fields whose lowered output was not finite; at the right edge, SNRs far above the floor, clipped there.



(b) ADAPT is run once per output field, with that field as its sole dependent, and the error bound it is given is swept. The axis is that bound $|\Delta J|$ as a fraction of the field's squared norm J, in decibels, not the field-variance $\mathrm{SNR}_{\mathrm{var}}$ of Equation (7). Each curve gives how many of the 55 registered variables ADAPT then refuses to demote from FP64. Six fields are shown; the cross on a curve marks the bound refinement derives for that field, refining the vertical levels and the substep together as Equation (12) requires.

Fig. 9. A 120 s CLOUDSC physics interval, taken in two substeps. Each of the 17 output fields for which a floor is defined sets its own threshold.

output. Those bounds span 24 dB across the 16 fields swept, and at them ADAPT leaves no arrays in FP64 for `PCOVPTOT` and `PFHPSN` but 20 for `PFSQLTUR`.

V. Related Work

Reduced Precision in Weather and Climate: Key atmospheric variables retain only 7–12 bits of real information [12], the ECMWF IFS runs in single precision with negligible forecast degradation [14], and the freed budget is better spent on ensemble size or resolution [11], [13]. These studies validate precision reduction *a posteriori*—through ensemble statistics or distributional comparison—but do not provide an automated, scalar criterion for a compiler-level tuning loop. Our work complements these findings with a principled threshold and a framework that automates the lowering.

Automated Precision Tuning: Dynamic approaches like PRECIMONIOUS [29] search for mixed-precision configurations satisfying user-defined error thresholds, but lack correctness guarantees for untested inputs. Static approaches like Daisy [35] compute worst-case rounding bounds via abstract interpretation or SMT solvers, but these bounds are often impractically pessimistic for long-running PDE simulations where errors cancel probabilistically [58]. Compiler-integrated rewriting goes further: Poseidon [59] instruments the program in a first pass, and runs it to capture numerical context. Then it uses that profile in a second pass to synthesize candidate rewrites, solving for a Pareto set of performance/accuracy tradeoffs. Emulators such as RAPTOR [21] take a format rather than a tolerance, leaving the verdict to the user. These tools measure error against a same-grid high-precision reference. Our framework evaluates against the physical discretization floor instead (Section II-D). Discretization-error-accurate multigrid [38] shares that baseline, but derives it analytically for SPD systems from linear elliptic PDEs. Ours is measured by refinement, so it applies wherever a refined run is available.

Sensitivity Analysis: Existing approaches use automatic differentiation [57] or variance decomposition [60] to compute per-variable error sensitivities. While effective for small kernels, instruction-level tracking across chaotic integration windows incurs prohibitive overhead. The adjoint estimate (Section IV-D) accumulates signed contributions, so round-off that cancels within the sum leaves it below the error actually incurred. Our framework sidesteps per-variable sensitivity, trading fine-grained optimality for scalability by validating aggregate output SNR against the discretization baseline.

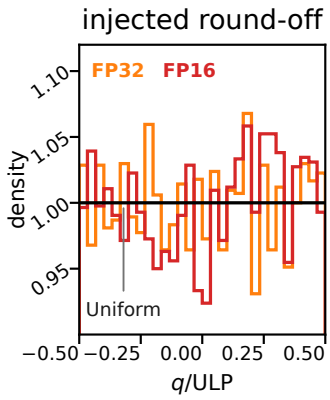
Probabilistic Error Analysis: Probabilistic bounds show that rounding errors in inner products grow as $\sqrt{n}$ rather than n [61], and Discrete Stochastic Arithmetic (DSA) estimates error propagation via randomized rounding [62]. However, DSA requires multiple executions or heavy instrumentation. Our framework shares the probabilistic premise but avoids repeated runs by evaluating single-execution variance directly against the discretization noise floor.

VI. Discussion and Future Work

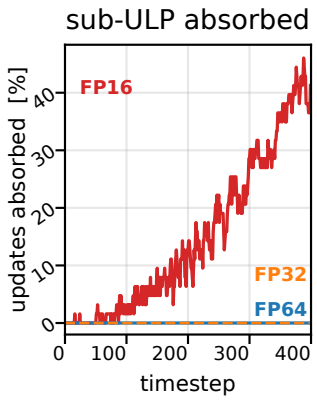
Our framework trades fine-grained optimality for scalability, and this choice has consequences worth discussing.

Uniform vs. mixed-precision granularity: Because we downcast uniformly, a single sensitive variable (e.g., `w_concorr_c` in ICON-VT) can block precision reduction for an entire subgraph. Extending the dataflow analysis to support intra-kernel precision mixing—using our coarse-grained SNR profiles to flag candidate regions for localized sensitivity analysis—would recover these missed opportunities.

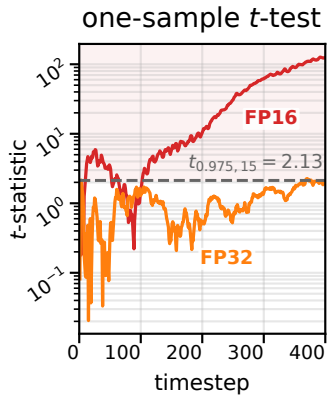
Hardware-dependent performance ceilings: When hardware lacks native support for specific low-precision operations (e.g., FP16 transcendentals), compiler-injected format conversions erode the bandwidth benefits of a smaller footprint [63]; FP16 VT kernel's marginal gain over FP32 despite halved



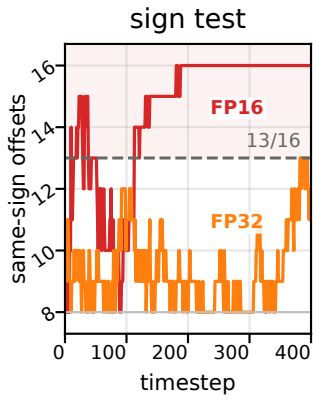
(a) q_p/ULP matches $\mathrm{Unif}(\pm\frac{1}{2})$: std 0.29, excess kurtosis -1.2; white, signal-independent ($\max_{k\geq 1}|\rho|, |\mathrm{corr}(q,u)| \leq 0.03$).



(b) FP16 absorbs sub-ULP updates (∼46% by late time); FP32/FP64 absorb none.



(c) t-test of $\overline{E_{ro}}$=0: FP32 stays below the 5% line, FP16 climbs to $t\sim 10^2$.



(d) Sign of $\overline{E_{ro}}$ across offsets: FP32 balanced (8/16), FP16 unanimous (16/16).

Fig. 10. Empirical test of the PQN assumptions (Burgers, FTCS). Round-off is $E_{ro} = u_p - u_{64}$, with spatial mean $\overline{E_{ro}}$. Panels (c, d) test whether this mean is real or a fluctuation: we re-solve at M=16 random sub-grid front offsets δ (same physics, independent round-off) and test $\overline{E_{ro}} = 0$ via $t = \sqrt{M}\,|\langle\overline{E_{ro}}\rangle_\delta|/\sigma_\delta$ ($\langle\cdot\rangle_\delta, \sigma_\delta$: mean, stddev over offsets). PQN holds at injection (a) but not under accumulation: the mean stays zero only while sub-ULP updates survive. FP16's signed absorption (b) drives a real bias (c, d); FP32 stays unbiased far longer, borderline only at the end.

memory traffic illustrates this ceiling. BF16 shows it more sharply: the same memory traffic as FP16, but division and square root lack native bfloat16 instructions and fall back to FP32 (Table IV). Emerging architectures with native BF16 datapaths shift this boundary; evaluating other representations such as block floating-point formats is a natural next step.

Limits of the PQN assumption: Treating round-off as uncorrelated pseudo-quantization noise [43] is justified for chaotic, macroscopic dynamics but breaks down when errors accumulate coherently or quantization noise swamps the physical update. Figure 10 illustrates this: the injected round-off satisfies the PQN assumptions (uniform, white, zero-mean), but once FP16's coarse ULP absorbs sub-ULP updates, the accumulated error develops a nonzero mean, while FP32 stays clean far longer. The empirical SNR checks guard against this, but only over the states and windows evaluated. Extending coverage to unseen regimes—longer horizons, broader ensembles, formal stability criteria—remains open.

Beyond weather and climate: Nothing in the criterion is specific to atmospheric models. It assumes only that the solver is discretization-limited, that a refined run can estimate the floor, and that the output is a fluctuating field whose variance is meaningful. Any stable solver run coarser than the resolution at which round-off would matter satisfies the first. It does not cover computation with no discretization to refine against; there, an arithmetic reference remains the right baseline.

VII. Conclusion

So, what *does* bloat your floats? In PDE-governed simulations, over half of the 52 significand bits drown in discretization noise—no amount of arithmetic precision can recover information the numerics never had. We formalize the intuition of "discretization noise floor" as a variance-based SNR, and encapsulate it in an automated DaCe-based framework that downcasts dataflow subgraphs and validates the results.

On ICON velocity tendencies and CLOUDSC, FP32 clears every baseline by a comfortable margin, translating to kernel speedups of up to 1.8× on both platforms—close to the theoretical 2× bandwidth ceiling. FP16 tells a more cautionary tale: viable in a sophisticated mixed-precision configuration (as it was for ICON-VT), and outright destructive when indiscriminately applied (as it was for CLOUDSC). BF16 does not clear the floor on either application. The safe precision boundary is application- and resolution-dependent, which is precisely why an automated criterion matters. Our framework trades per-variable optimality for scalability, the alternative being an unscalable combinatorial search. Finer-grained mixed precision, broader input coverage, and emerging block formats are natural next steps.

The bits discretization renders irrelevant are headroom better spent on finer grids, higher-order schemes, or larger ensembles—turning wasted arithmetic into actual science.

Acknowledgment

This project received support from the ERC PSAP project (Grant Agreement No. 101002047). We thank the Swiss National Supercomputing Centre (CSCS) for providing computational resources and support that facilitated this project.

This work used AI assistants—Claude [64], Gemini [65], and Overleaf AI [66]—for code development, language editing, typesetting, and bibliographic cross-referencing throughout. All outputs were verified by the authors.

References

[1] S. Bhola and K. Duraisamy, *Deterministic and probabilistic rounding error analysis for mixed-precision arithmetic on modern computing units*, Nov. 27, 2024. DOI: 10.48550/arXiv.2411.18747.

[2] N. J. Higham and T. Mary, "Mixed precision algorithms in numerical linear algebra," *Acta Numerica*, vol. 31, pp. 347–414, May 2022, ISSN: 0962-4929, 1474-0508. DOI: 10.1017/S0962492922000022.

[3] N. J. Higham, *Accuracy and stability of numerical algorithms*, 2nd ed. Philadelphia: Society for Industrial and Applied Mathematics, 2002, 680 pp., ISBN: 978-0-89871-521-7. DOI: 10.1137/1.9780898718027.

[4] L. N. Trefethen and D. Bau III, *Numerical Linear Algebra*. SIAM, Jun. 1, 1997, 356 pp., ISBN: 978-0-89871-361-9. DOI: 10.1137/1.9780898719574.

[5] J. Dongarra, J. Gunnels, H. Bayraktar, A. Haidar, and D. Ernst, *Hardware trends impacting floating-point computations in scientific applications*, Dec. 20, 2024. DOI: 10.48550/arXiv.2411.12090.

[6] NVIDIA, "NVIDIA blackwell architecture technical brief," NVIDIA, 2024. Accessed: Aug. 25, 2026. [Online]. Available: https://www.nvidia.com/en-us/data-center/technologies/blackwell-architecture/.

[7] A. Walden, E. Nielsen, B. Diskin, and M. Zubair, "A mixed precision multicolor point-implicit solver for unstructured grids on GPUs," in *2019 IEEE/ACM 9th Workshop on Irregular Applications: Architectures and Algorithms (IA3)*, Nov. 2019, pp. 23–30. DOI: 10.1109/IA349570.2019.00010.

[8] D. Unat, A. Dubey, E. Jeannot, and J. Shalf, "The persistent challenge of data locality in the post-exascale era," *Computing in Science & Engineering*, vol. 27, no. 4, pp. 19–27, Oct. 2025, ISSN: 1558-366X. DOI: 10.1109/MCSE.2025.3567586.

[9] O. Fuhrer et al., "Near-global climate simulation at 1 km resolution: Establishing a performance baseline on 4888 GPUs with COSMO 5.0," *Geoscientific Model Development*, vol. 11, no. 4, pp. 1665–1681, May 2, 2018, ISSN: 1991-959X. DOI: 10.5194/gmd-11-1665-2018.

[10] J. Shalf, "The future of computing beyond moore's law," *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 378, no. 2166, p. 20 190 061, Jan. 20, 2020. DOI: 10.1098/rsta.2019.0061.

[11] M. Karp et al., *Effects of lower floating-point precision on scale-resolving numerical simulations of turbulence*, Jun. 5, 2025. DOI: 10.48550/arXiv.2506.05150.

[12] M. Klöwer, "Low-precision climate computing: Preserving information despite fewer bits," Ph.D. dissertation, University of Oxford, 2021. Accessed: Mar. 16, 2026. [Online]. Available: https://ora.ox.ac.uk/objects/uuid:1158e44a-7faf-45a0-8ab1-73c91fd694a6.

[13] T. N. Palmer, "More reliable forecasts with less precise computations: A fast-track route to cloud-resolved weather and climate simulators?" *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 372, no. 2018, p. 20 130 391, Jun. 28, 2014, ISSN: 1364-503X. DOI: 10.1098/rsta.2013.0391.

[14] F. Váňa et al., "Single precision in weather forecasting models: An evaluation with the IFS," *Monthly Weather Review*, vol. 145, no. 2, pp. 495–502, Feb. 1, 2017, ISSN: 1520-0493, 0027-0644. DOI: 10.1175/MWR-D-16-0228.1.

[15] T. Thornes, P. Düben, and T. Palmer, "A power law for reduced precision at small spatial scales: Experiments with an SQG model," *Quarterly Journal of the Royal Meteorological Society*, vol. 144, no. 713, pp. 1179–1188, 2018, ISSN: 1477-870X. DOI: 10.1002/qj.3303.

[16] W. Weens, T. Vazquez-Gonzalez, and L. B. Salem-Knapp, "Modeling round-off errors in hydrodynamic simulations," in *Software Verification*, R. Bloem, R. Dimitrova, C. Fan, and N. Sharygina, Eds., Cham: Springer International Publishing, 2022, pp. 182–196, ISBN: 978-3-030-95561-8. DOI: 10.1007/978-3-030-95561-8_11.

[17] T. N. Palmer, "Towards the probabilistic earth-system simulator: A vision for the future of climate and weather prediction," *Quarterly Journal of the Royal Meteorological Society*, vol. 138, no. 665, pp. 841–861, 2012, ISSN: 1477-870X. DOI: 10.1002/qj.1923.

[18] D. Bauer, N. Kohl, S. F. McCormick, and R. Tamstorf, *Multigrid with linear storage complexity*, Nov. 24, 2025. DOI: 10.48550/arXiv.2511.19036.

[19] N. Demeure, C. Chevalier, C. Denis, and P. Dossantos-Uzarralde, "Algorithm 1029: Encapsulated error, a direct approach to evaluate floating-point accuracy," *ACM Transactions on Mathematical Software*, vol. 48, no. 4, 47:1–47:16, Mar. 22, 2023, ISSN: 0098-3500. DOI: 10.1145/3549205.

[20] J. D. Vanover, "Engineering floating-point correctness: Practical testing and tuning techniques," Ph.D. dissertation, UC Davis, Sep. 21, 2025. Accessed: Mar. 24, 2026. [Online]. Available: https://escholarship.org/uc/item/8n1443pk.

[21] F. Hoerold et al., "RAPTOR: Practical numerical profiling of scientific applications," in *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, St. Louis MO USA: ACM, Nov. 16, 2025, pp. 661–680, ISBN: 979-8-4007-1466-5. DOI: 10.1145/3712285.3759810.

[22] P. Bauer, A. Thorpe, and G. Brunet, "The quiet revolution of numerical weather prediction," *Nature*, vol. 525, no. 7567, pp. 47–55, Sep. 2015, ISSN: 1476-4687. DOI: 10.1038/nature14956.

[23] P. Bauer, P. D. Dueben, T. Hoefler, T. Quintino, T. C. Schulthess, and N. P. Wedi, "The digital revolution of earth-system science," *Nature Computational Science*, vol. 1, no. 2, pp. 104–113, Feb. 2021, ISSN: 2662-8457. DOI: 10.1038/s43588-021-00023-0.

[24] S. Matsuoka, J. Domke, M. Wahib, A. Drozd, and T. Hoefler, "Myths and legends in high-performance computing," *The International Journal of High Performance Computing Applications*, vol. 37, no. 3, pp. 245–259,

Jul. 2023, ISSN: 1094-3420, 1741-2846. DOI: 10.1177/10943420231166608.

[25] F. Benz, A. Hildebrandt, and S. Hack, "A dynamic program analysis to find floating-point accuracy problems," *SIGPLAN Not.*, vol. 47, no. 6, pp. 453–462, Jun. 11, 2012, ISSN: 0362-1340. DOI: 10.1145/2345156.2254118.

[26] A. Kashi et al., *Mixed-precision numerics in scientific applications: Survey and perspectives*, Dec. 26, 2024. DOI: 10.48550/arXiv.2412.19322.

[27] S. Graillat, F. Jézéquel, R. Picot, F. Févotte, and B. Lathuilière, "Auto-tuning for floating-point precision with discrete stochastic arithmetic," *Journal of Computational Science*, vol. 36, p. 101 017, Sep. 1, 2019, ISSN: 1877-7503. DOI: 10.1016/j.jocs.2019.07.004.

[28] G. Constantinides, F. Dahlqvist, Z. Rakamarić, and R. Salvia, "Automated roundoff error analysis of probabilistic floating-point computations," *ACM Trans. Probab. Mach. Learn.*, vol. 1, no. 3, 13:1–13:35, Aug. 26, 2025. DOI: 10.1145/3705898.

[29] C. Rubio-González et al., "Precimonious: Tuning assistant for floating-point precision," in *Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis*, ser. SC '13, New York, NY, USA: Association for Computing Machinery, Nov. 17, 2013, pp. 1–12, ISBN: 978-1-4503-2378-9. DOI: 10.1145/2503210.2503296.

[30] H. Wan et al., "The ICON-1.2 hydrostatic atmospheric dynamical core on triangular grids – part 1: Formulation and performance of the baseline version," *Geoscientific Model Development*, vol. 6, no. 3, pp. 735–763, 2013. DOI: 10.5194/gmd-6-735-2013.

[31] G. Zängl, D. Reinert, P. Rípodas, and M. Baldauf, "The ICON (ICOsahedral non-hydrostatic) modelling framework of DWD and MPI-m: Description of the non-hydrostatic dynamical core," *Quarterly Journal of the Royal Meteorological Society*, vol. 141, no. 687, pp. 563–579, 2015. DOI: 10.1002/qj.2378.

[32] C. Jablonowski and D. L. Williamson, "A baroclinic instability test case for atmospheric model dynamical cores," *Quarterly Journal of the Royal Meteorological Society*, vol. 132, no. 621, pp. 2943–2975, 2006, ISSN: 1477-870X. DOI: 10.1256/qj.06.12.

[33] K. Course and P. B. Nair, "State estimation of a physical system with unknown governing equations," *Nature*, vol. 622, no. 7982, pp. 261–267, Oct. 2023, ISSN: 1476-4687. DOI: 10.1038/s41586-023-06574-8.

[34] L. Saffin, S. Hatfield, P. Düben, and T. Palmer, "Reduced-precision parametrization: Lessons from an intermediate-complexity atmospheric model," *Quarterly Journal of the Royal Meteorological Society*, vol. 146, no. 729, pp. 1590–1607, Apr. 2020, ISSN: 0035-9009, 1477-870X. DOI: 10.1002/qj.3754.

[35] E. Darulova, A. Izycheva, F. Nasir, F. Ritter, H. Becker, and R. Bastian, "Daisy - framework for analysis and optimization of numerical programs (tool paper)," in *Tools and Algorithms for the Construction and Analysis of Systems*, D. Beyer and M. Huisman, Eds., Cham: Springer International Publishing, 2018, pp. 270–287, ISBN: 978-3-319-89960-2. DOI: 10.1007/978-3-319-89960-2_15.

[36] ECMWF, *Ecmwf-ifs/dwarf-p-cloudsc*, Sep. 4, 2025. Accessed: Aug. 19, 2026. [Online]. Available: https://github.com/ecmwf-ifs/dwarf-p-cloudsc.

[37] R. J. LeVeque, *Finite Difference Methods for Ordinary and Partial Differential Equations: Steady-State and Time-dependent Problems*. Philadelphia, PA: Society for Industrial and Applied Mathematics, 2007, 184 pp., ISBN: 978-0-89871-629-0. DOI: 10.1137/1.9780898717839.

[38] R. Tamstorf, J. Benzaken, and S. F. McCormick, "Discretization-error-accurate mixed-precision multigrid solvers," *SIAM Journal on Scientific Computing*, vol. 43, no. 5, S420–S447, Jan. 2021, ISSN: 1064-8275. DOI: 10.1137/20M1349230.

[39] A. Izycheva and E. Darulova, "On sound relative error bounds for floating-point arithmetic," in *2017 Formal Methods in Computer Aided Design (FMCAD)*, Oct. 2017, pp. 15–22. DOI: 10.23919/FMCAD.2017.8102236.

[40] G. Peyré and M. Cuturi, *Computational optimal transport*, Mar. 18, 2020. DOI: 10.48550/arXiv.1803.00567.

[41] A. H. Baker, A. Pinard, and D. M. Hammerling, *DSSIM: A structural similarity index for floating-point data*, Mar. 19, 2023. DOI: 10.48550/arXiv.2202.02616.

[42] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE transactions on image processing*, vol. 13, no. 4, pp. 600–612, 2004. DOI: 10.1109/TIP.2003.819861.

[43] B. Widrow and I. Kollár, *Quantization Noise: Roundoff Error in Digital Computation, Signal Processing, Control, and Communications*. Cambridge: Cambridge University Press, 2008, ISBN: 978-0-521-88671-0. DOI: 10.1017/CBO9780511754661.

[44] T. E. Hull and J. R. Swenson, "Tests of probabilistic models for propagation of roundoff errors," *Commun. ACM*, vol. 9, no. 2, pp. 108–113, Feb. 1, 1966, ISSN: 0001-0782. DOI: 10.1145/365170.365212.

[45] K. Kalliojarvi and J. Astola, "Roundoff errors in block-floating-point systems," *IEEE Transactions on Signal Processing*, vol. 44, no. 4, pp. 783–790, Apr. 1996, ISSN: 1941-0476. DOI: 10.1109/78.492531.

[46] A. Dalcher and E. Kalnay, "Error growth and predictability in operational ECMWF forecasts," *Tellus A: Dynamic Meteorology and Oceanography*, vol. 39, no. 5, p. 474, Jan. 1, 1987, ISSN: 1600-0870. DOI: 10.3402/tellusa.v39i5.11774.

[47] H. Savijarvi, "Error growth in a large numerical forecast system," *Monthly weather review*, vol. 123, no. 1, pp. 212–221, 1995. DOI: 10.1175/1520-0493(1995)123⟨0212:EGIALN⟩2.0.CO;2.

[48] M. P. Bonkile, A. Awasthi, C. Lakshmi, V. Mukundan, and V. S. Aswin, "A systematic literature review of burgers' equation with recent advances," *Pramana*, vol. 90, no. 6, p. 69, Apr. 30, 2018, ISSN: 0973-7111. DOI: 10.1007/s12043-018-1559-4.

[49] A. N. Kolmogorov and V. Levin, "The local structure of turbulence in incompressible viscous fluid for very large reynolds numbers," *Proceedings of the Royal Society of London. Series A: Mathematical and Physical Sciences*, vol. 434, no. 1890, J. C. R. Hunt, O. M. Phillips, and D. Williams, Eds., pp. 9–13, Jul. 8, 1991, ISSN: 0962-8444. DOI: 10.1098/rspa.1991.0075.

[50] G. D. Nastrom and K. S. Gage, "A climatology of atmospheric wavenumber spectra of wind and temperature observed by commercial aircraft," *Journal of the Atmospheric Sciences*, vol. 42, no. 9, pp. 950–960, May 1, 1985, ISSN: 0022-4928, 1520-0469. DOI: 10.1175/1520-0469(1985)042⟨0950:ACOAWS⟩2.0.CO;2.

[51] W. C. Skamarock, "Evaluating mesoscale NWP models using kinetic energy spectra," *Monthly Weather Review*, vol. 132, no. 12, pp. 3019–3032, Dec. 1, 2004, ISSN: 1520-0493, 0027-0644. DOI: 10.1175/MWR2830.1.

[52] T. Ben-Nun, J. d. F. Licht, A. N. Ziogas, T. Schneider, and T. Hoefler, *Stateful dataflow multigraphs: A data-centric model for performance portability on heterogeneous architectures*, Jan. 2, 2020. DOI: 10.48550/arXiv.1902.10345.

[53] D. Klocke et al., "Computing the full earth system at 1km resolution," in *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, ser. SC '25, New York, NY, USA: Association for Computing Machinery, Nov. 15, 2025, pp. 125–136, ISBN: 979-8-4007-1466-5. DOI: 10.1145/3712285.3771789.

[54] L. Fusco, M. Khalilov, M. Chrapek, G. Chukkapalli, T. Schulthess, and T. Hoefler, *Understanding data movement in tightly coupled heterogeneous systems: A case study with the grace hopper superchip*, Aug. 26, 2024. DOI: 10.48550/arXiv.2408.11556.

[55] T. Hoefler and R. Belli, "Scientific benchmarking of parallel computing systems: Twelve ways to tell the masses when reporting performance results," in *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, Austin Texas: ACM, Nov. 15, 2015, pp. 1–12, ISBN: 978-1-4503-3723-6. DOI: 10.1145/2807591.2807644.

[56] NVIDIA, "Matrix multiplication background user's guide," NVIDIA, 2023. Accessed: Aug. 19, 2026. [Online]. Available: https://docs.nvidia.com/deeplearning/performance/dl-performance-matrix-multiplication/index.html.

[57] H. Menon et al., "ADAPT: Algorithmic differentiation applied to floating-point precision tuning," in *Proceedings of the International Conference for High Performance Computing, Networking, Storage, and Analysis*, ser. SC '18, Dallas, Texas: IEEE Press, Jul. 26, 2019, pp. 1–13. DOI: 10.1109/SC.2018.00051.

[58] A. Isychev and D. Lohar, "Cost of soundness in mixed-precision tuning," *Proceedings of the ACM on Programming Languages*, vol. 9, pp. 2426–2453, OOPSLA2 Oct. 9, 2025, ISSN: 2475-1421. DOI: 10.1145/3763137.

[59] S. B. Qian, V. Sathia, I. R. Ivanov, J. Hückelheim, P. Hovland, and W. S. Moses, "Thinking fast and correct: Automated rewriting of numerical code through compiler augmentation," in *2026 IEEE/ACM International Symposium on Code Generation and Optimization (CGO)*, Jan. 2026, pp. 548–562. DOI: 10.1109/CGO68049.2026.11395228.

[60] M. Cui, J. Xu, Y. Zhou, H. Yang, L. Ji, and B. Zhou, "PESA: Error sensitivity analysis tool for floating-point computational programs," *The Journal of Supercomputing*, vol. 81, no. 3, p. 477, Feb. 10, 2025, ISSN: 1573-0484. DOI: 10.1007/s11227-025-06962-z.

[61] N. J. Higham and T. Mary, "A new approach to probabilistic rounding error analysis," *SIAM Journal on Scientific Computing*, Sep. 12, 2019. DOI: 10.1137/18M1226312.

[62] F. Jézéquel and J.-M. Chesneaux, "CADNA: A library for estimating round-off error propagation," *Computer Physics Communications*, vol. 178, no. 12, pp. 933–955, Jun. 15, 2008, ISSN: 0010-4655. DOI: 10.1016/j.cpc.2008.02.003.

[63] S. Heldens and B. van Werkhoven, "Kernel float: Unlocking mixed-precision GPU programming," *ACM Trans. Math. Softw.*, vol. 52, no. 1, 2:1–2:36, Mar. 17, 2026, ISSN: 0098-3500. DOI: 10.1145/3779120.

[64] Anthropic, "The claude 4.6 model family: Opus, sonnet, and haiku technical report," Anthropic, Feb. 2026. Accessed: Aug. 25, 2026. [Online]. Available: https://www-cdn.anthropic.com/0dd865075ad3132672ee0ab40b05a53f14cf5288.pdf.

[65] G. Team et al., *Gemini: A family of highly capable multimodal models*, May 9, 2025. DOI: 10.48550/arXiv.2312.11805.

[66] Overleaf, *Overleaf AI assist (including writefull and GPT-powered tools) [large language model]*, 2026. Accessed: Aug. 25, 2026. [Online]. Available: https://www.overleaf.com/about/ai-features.