Solar Neutrino Flux Predictions with Terrestrial Matter Effects

— Sun & Earth Adaptive Simulation Of Neutrinos

Keyu Han
Tsinghua University

Motivation

As future solar-neutrino experiments become increasingly precise, numerical predictions may also need to reach the sub-percent level.

At this precision, several previously subleading effects must be treated systematically:

- Introduce the 3D Earth-density model into terrestrial matter effects
- High-precision astrometry model to determine the Earth-Sun positions varied with time
- finite-size effects of the ^{8}B solar production region
- propagation of oscillation-parameter uncertainties through the full calculation chain.

However, these refinements greatly increase the computational cost.
A brute-force treatment of all solar-source points, Earth-crossing trajectories, energy grids, and Monte Carlo samples becomes impractical.

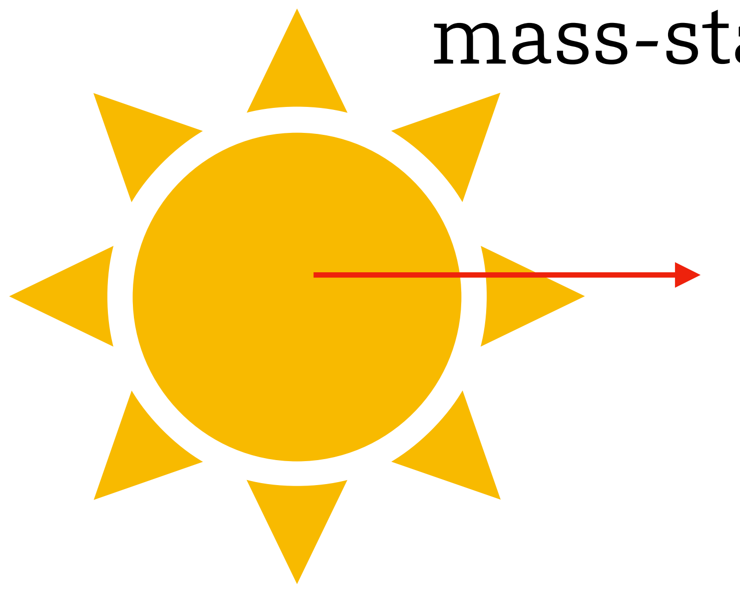
Goal of this Framework: build a fast end-to-end Sun–Earth simulation framework that keeps realistic physics inputs while remaining efficient enough for site-by-site predictions and uncertainty propagation.

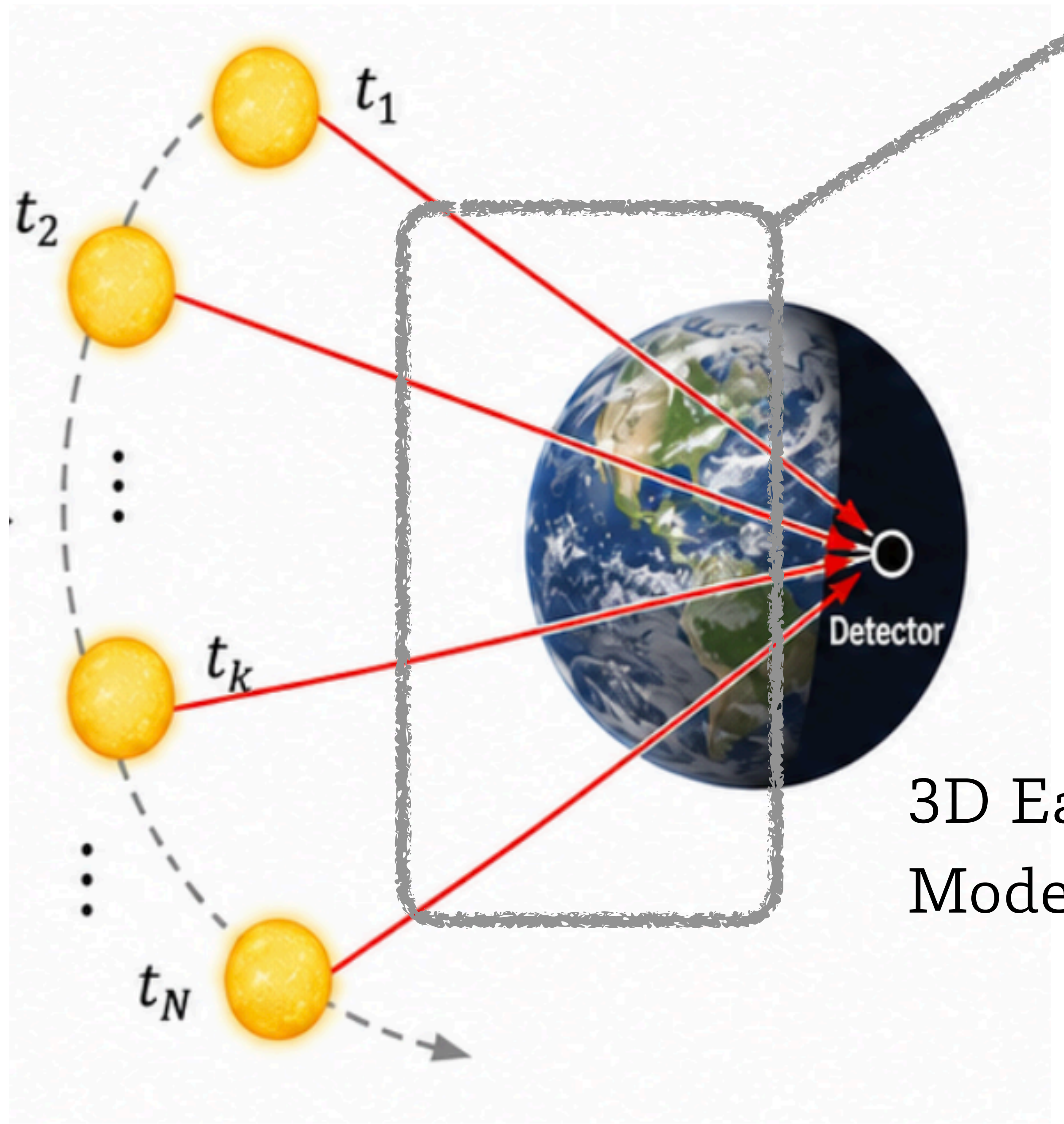
Framework Overview

Solar-side propagation

P_{ei}

mass-state composition



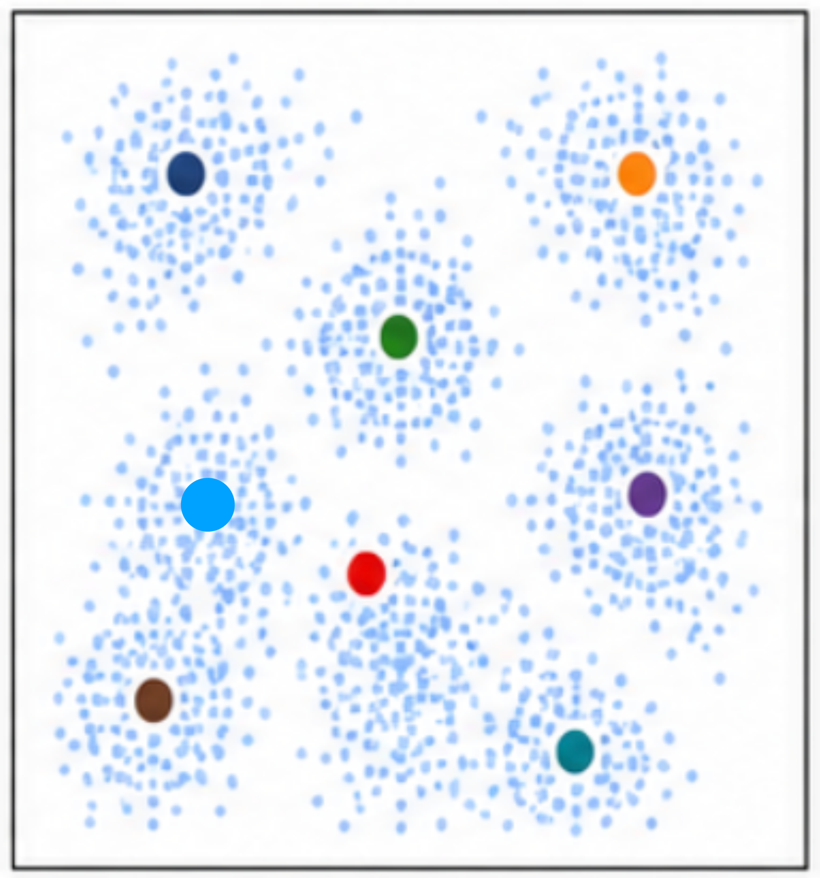


3D Earth Density Model

Time-dependent Sun–Earth geometry

Representative path clustering

compress many similar trajectories into a smaller set of representative paths



Earth-side propagation

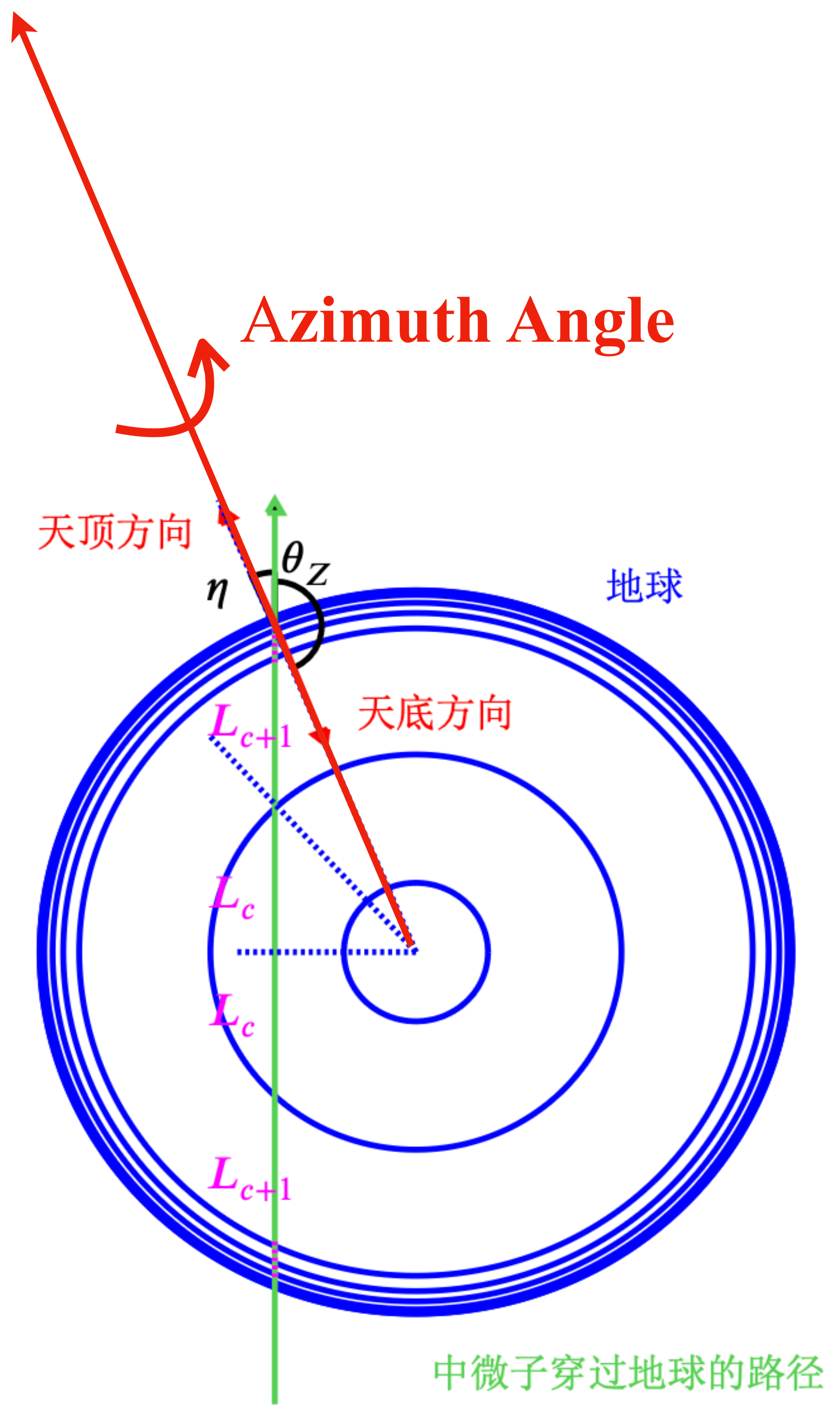
Why 3D Earth Makes the Problem Hard?

From 1D to 3D Earth: Trajectory Space Explosion

In a spherically symmetric Earth model, the terrestrial matter effect depends mainly on the nadir angle η.

With a 3D Earth-density model, the azimuth angle also becomes relevant:

$$\text{trajectory} \sim (\eta, \text{az})$$



A full-year time sampling with minute-level resolution produces hundreds of thousands of Sun–Earth geometries.

For nighttime samples alone, this gives approximately $N_{\text{night}} \simeq 2.6 \times 10^5 (1\ min)$ Earth-crossing trajectories from Sun-centers before considering the finite solar production region.

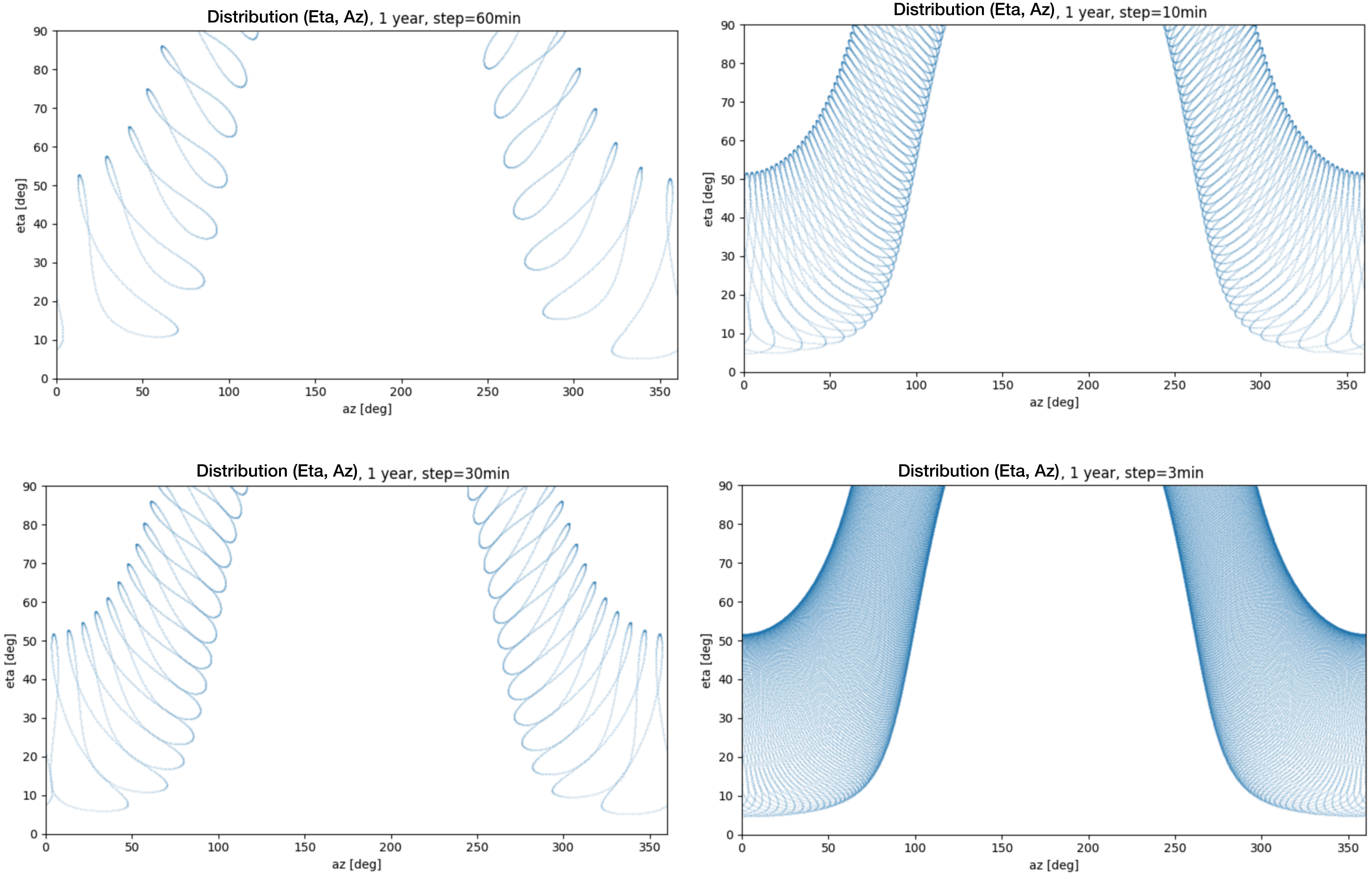
Why 3D Earth Makes the Problem Hard?

—Why Minute-Level Time Sampling Is Needed ?

The annual Sun–Earth geometry forms a continuous structure in this two-dimensional trajectory space.

A coarse time step can undersample this structure and miss important regions of the nighttime path distribution.

Minute-level sampling is therefore required to obtain a dense and unbiased set of central Earth-crossing trajectories.



Finite ^{8}B Source and Path Clustering

The $\sim 2.6\times 10^{5}$ nighttime trajectories discussed above correspond only to the **central Sun direction**.

A more realistic calculation should include the finite ^{8}B solar production region, rather than treating the source as a single point at the solar center. Using the 1D Bahcall ^{8}B production profile, a straightforward treatment typically requires at least $\sim \mathcal{O}(10^{2}-10^{3})$ weighted nodes or MC samples per time step.

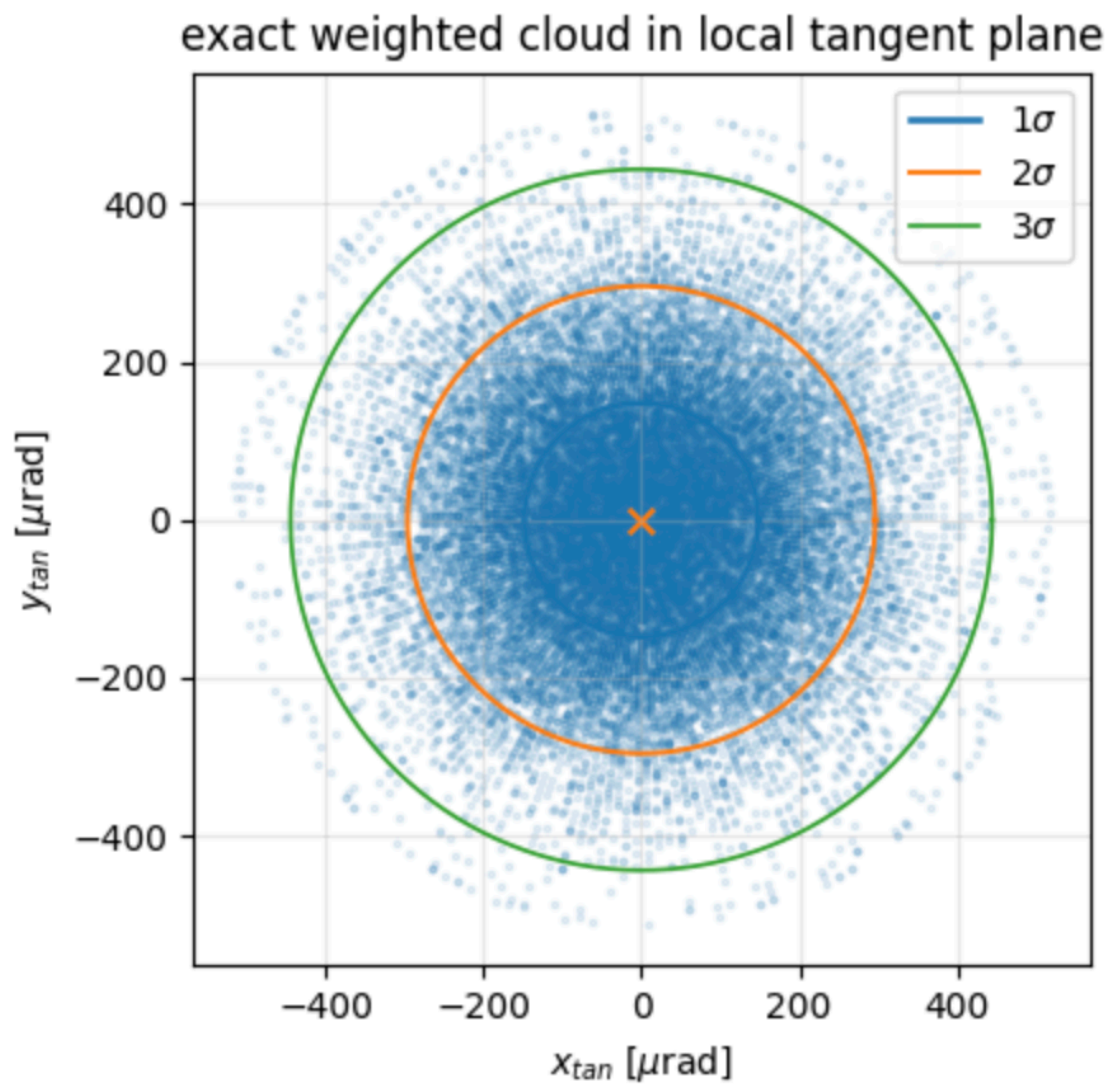
We first represent each nighttime trajectory (η, az) as a unit vector(detector point to Sun center, line-of-sight):

$$(\eta, az) \;\longrightarrow\; \mathbf{u}(\eta, az) = \begin{pmatrix} \sin\eta\ \sin az \\ \sin\eta\ \cos az \\ -\cos\eta \end{pmatrix}.$$

With the detector as the origin, these unit vectors lie on the unit sphere.

Viewed from the detector, each source point corresponds to a slightly different incoming unit vector on the unit sphere.

- Around the central Sun direction, these unit vectors form a small angular point cloud.
- In the local tangent plane, this cloud is nearly isotropic and can be summarized by a center direction and an angular width.
- Empirically, these point clouds are well approximated by 2D isotropic Gaussian clouds.



This motivates clustering **finite-source angular clouds**, rather than clustering single central rays.

Unsupervised Machine Learning: Neutrino Path Clustering

For two Gaussian clouds on a Euclidean plane,

$$\mathcal{N}(\mathbf{m}_i, \Sigma_i), \qquad \mathcal{N}(\mathbf{m}_j, \Sigma_j),$$

the Wasserstein-2 distance has a closed-form expression. For isotropic 2D Gaussian clouds, ($\Sigma_i = \sigma_i^2 I$), it reduces to

$$W_{ij}^2 = |\mathbf{m}_i - \mathbf{m}_j|^2 + 2(\sigma_i - \sigma_j)^2 .$$

For our finite-source clouds live on the unit sphere, so we replace the Euclidean center displacement by the geodesic angular distance, we call it W2-inspired surrogate distance:

$$\mathcal{W}_{ij}^2 = \theta_{ij}^2 + 2\left(\sigma_{\tan,i} - \sigma_{\tan,j}\right)^2, \qquad \theta_{ij} = \arccos(\mathbf{u}_i \cdot \mathbf{u}_j),$$

where lower index 'tan' denotes the local tangent-plane, $\mathbf{u}_i$ is the center of the point clouds.

K-Medoids Clustering procedure

1. Initialize K representative samples from the nighttime set.
2. Assign each cloud to the nearest representative using the W2-inspired surrogate distance.
3. Update each cluster by selecting a real sample as its representative.
4. Iterate until stable.

Unsupervised Machine Learning: Neutrino Path Clustering

- Left: dense nighttime trajectory distribution in the (η, az) space.
- Right: the same samples colored by cluster label for (K=2000).

Each cluster is represented by a real nighttime trajectory and later used for Earth-side MSW propagation.

Earth-Side Propagation: Strang-splitting

For each representative Earth-crossing trajectory, the exact evolution operator along one trajectory is a path-ordered exponential,

$$S^{\oplus}(E) = \mathscr{P}exp[-i\int_0^{L_\oplus} H(x;E)dx].$$

with

$$H(x;E) = H_{vac}(E) + H_{mat}(x), \qquad H_{\text{vac}} = \frac{1}{2E}\begin{pmatrix} 0 & 0 & 0 \\ 0 & \Delta m_{21}^2 & 0 \\ 0 & 0 & \Delta m_{31}^2 \end{pmatrix} \qquad H_{\text{mat}} = U^\dagger \begin{pmatrix} V_e & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} U$$

We discretize the trajectory into matter layers and evaluate the matter potential at each layer midpoint:

$$\begin{aligned} S^{\oplus}(E;\ x_N = L_\oplus, 0) &\approx S_{N-1}(E)\ldots S_1(E)S_0(E) \\ &= e^{-iH_{N-1}\Delta x_{N-1}} \cdots e^{-iH_1\Delta x_1} e^{-iH_0\Delta x_0}. \end{aligned}$$

Within each layer propagator, using symmetric Strang splitting,

$$S_n(E) \simeq e^{-iH_{vac}\frac{\Delta x}{2}}\, e^{-iH_{mat}\Delta x}\, e^{-iH_{vac}\frac{\Delta x}{2}} + O(\Delta x_n^3).$$

Earth-Side Propagation: Strang-splitting

For the symmetric Strang splitting form,

$$S_n(E) \simeq e^{-iH_{vac}\frac{\Delta x}{2}}\, e^{-iH_{mat}\Delta x}\, e^{-iH_{vac}\frac{\Delta x}{2}}$$

In the vacuum mass basis, the matter Hamiltonian has the form,

$$H_{\rm mat} = U^\dagger \begin{pmatrix} V_n & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} U = V_n U^\dagger \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \end{pmatrix} U,$$

We define

$$|\,e\rangle = U^\dagger \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} = (U^*_{e1}, U^*_{e2}, U^*_{e3})^T, \qquad P_e = |\,e\rangle\langle e\,|\,.$$

The matter exponential is evaluated exactly as

$$\begin{aligned} e^{-iV_n\Delta x_n P_e} &= I + \sum_{n=1} \frac{1}{n!}(-iV_k\Delta x_k)^n P_e^n \\ &= I + \left(e^{-iV_k\Delta x_k} - 1\right) P_e\,. \end{aligned}$$

Therefore, the middle Strang step is only a rank-one update:

$$|\psi_{n+1}\rangle \;=\; e^{-iH_{\rm vac}\Delta x_n/2}\left[I + \left(e^{-iV_n\Delta x_n} - 1\right) P_e\right] e^{-iH_{\rm vac}\Delta x_n/2}\,|\psi_n\rangle\,.$$

Earth-Side Propagation: Strang-splitting

Why Strang splitting is useful here $$|\psi_{n+1}\rangle = e^{-iH_{\mathrm{vac}}\Delta x_n/2}\left[I + \left(e^{-iV_n\Delta x_n} - 1\right) P_e\right] e^{-iH_{\mathrm{vac}}\Delta x_n/2}\,|\psi_n\rangle\,.$$

- H_vac is diagonal in the mass basis:
the vacuum half-steps are just phase multiplications.

- H_mat is a rank-one projector:
the matter step is an exact closed-form rank-one update.

- P_e and H_{vac} can always be pre-computed before propagations

- No generic matrix exponential and repeated 3x3 eigensolver is needed:
avoids calling expm or eigh for every layer, energy, and path.

- The computation becomes a fixed sequence of scalar operations:
well suited for GPU parallelization over paths and energies.

Method	Issue for large path-energy scans
expm/eigh	Accurate, but need iterations, expensive
RK4/RKF	Not exactly unitary; dealing with large $N_E \times N_{layer}$ scans
Strang	Closed-form update; separates the **energy-dependent** and **layer-dependent** parts

Solar-Side Propagation: Adaptive CF4 Evolution

Why a Different Strategy Is Needed ?

- **The propagation scale is much longer. The solar radius is about $R_\odot \simeq 110\, R_\oplus \simeq 55 D_\oplus$.**

  For the discretization $S^\odot(E;\ x_N = L_\odot,\ 0) \approx e^{-iH_{N-1}\Delta x_{N-1}} \cdots e^{-iH_1\Delta x_1} e^{-iH_0\Delta x_0}$, this demands significant numbers of layer steps .

  (i.e. If $N_{layer,\oplus} = 8000$, then $N_{layer,\odot} \sim 440000$)

- **The solar-core electron density and matter potential are much larger than those in the Earth**

  Matrices splitting methods such as Strang-splitting introduce BCH commutator errors $\sim \mathcal{O}\left(\Delta x^3 [H_{\rm mat}, [H_{\rm mat}, H_{\rm vac}]] + \Delta x^3 [H_{\rm vac}, [H_{\rm vac}, H_{\rm mat}]]\right)$, which become difficult to suppress when the matter potential is large and rapidly varying.

Therefore, for the solar side we use a different strategy:

high-order CF4 propagation $+$ closed-form stage exponentials $+$ adaptive mesh construction .

Solar-Side Propagation: Adaptive CF4 Evolution

- The propagation scale is much longer.

The exact evolution operator along one trajectory is a path-ordered exponential,

$$S^{\odot}(E) = \mathscr{P}exp[-i\int_0^{L_\odot} H(x;E)dx].$$

We adopt a 4th-order commutator-free Magnus expansion,

$$S_{\mathrm{CF4}}(E_k, \Delta x_n) = \exp\left[-i\Delta x_n(a_1 H_1 + a_2 H_2)\right] \exp\left[-i\Delta x_n(a_2 H_1 + a_1 H_2)\right],$$

where H_1 and H_2 are evaluated at two internal sampling points inside the step.

$$H_i = H(x_n + c_i \Delta x_n; E_k), \qquad a_{1,2} = \frac{1}{4} \pm \frac{\sqrt{3}}{6}, \qquad c_{1,2} = \frac{1}{2} \mp \frac{\sqrt{3}}{6}\,.$$

Magnus CF4 expansion allows us to use larger layer steps to get the target accuracy.

Solar-Side Propagation: Adaptive CF4 Evolution

The solar-core electron density and matter potential are much larger—

—but smooth, nearly adiabatic in the LMA region.
In the LMA region, no severe near-degeneracy appears, so the eigenvalues can be stably obtained from a cubic equation by cubic discriminant. The stage exponential is then evaluated with the Cayley-Hamilton form during the full numerical propagation:

$$exp(-iT\Delta x) = \alpha_0 I + \alpha_1 T + \alpha_2 T^2$$

where CH parameters α_0, α_1 and α_2 are determined by the eigenvalues of the matrix T.

Solar-Side Propagation: Cheap 2×2 Predictor

The purpose of adaptive step construction is not to compute the final probability directly.

For any adaptive method, the basic idea is to compare **a local error** indicator with **a local tolerance**:
- if the local error is small, the step is accepted;
- if the local error is large, the step is refined.

The local tolerance itself is not an absolute physical observable.
It must be calibrated by convergence tests against the final target accuracy.

Therefore, during mesh construction, we do not need to use the full 3×3 solar propagation every time.

Instead, we use a cheap 2×2 active-block predictor to **capture the dominant MSW behavior in the solar interior.**

The predictor only decides where the adaptive mesh should be refined.

The final solar propagation is still performed with the full 3×3 CF4 evolution, and the predictor tolerance is calibrated by full 3×3 convergence tests.

Key point:
The 2×2 predictor builds the mesh; the full 3×3 propagation gives the physics result.

^8B Neutrino Result of CJPL

The flux and the asymmetry with their MC spread are given below (PDG parameters uncertainties and Bahcall energy spectrum uncertainties),
where the asymmetry is determined only from the day-night flux difference, no cross-section considered.

Phi_day: $2.037^{+0.0296}_{-0.0258} \times 10^6 cm^{-2}s^{-1}$,

Phi_night: $2.073^{+0.0301}_{-0.0258} \times 10^6 cm^{-2}s^{-1}$,

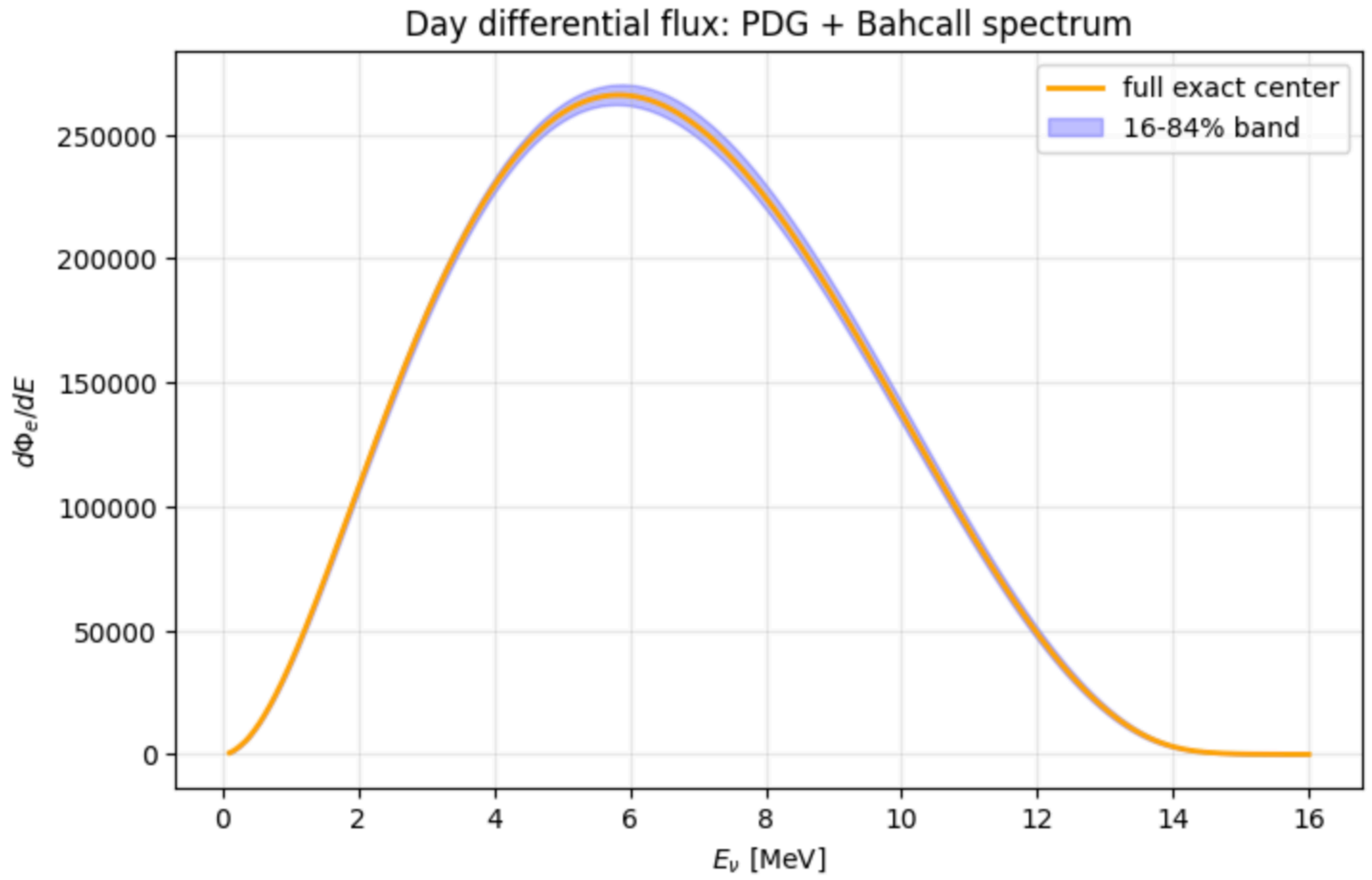
A_DN: $1.75^{+0.078}_{-0.068}\,\%$, $[0.1,\ 16] MeV$,

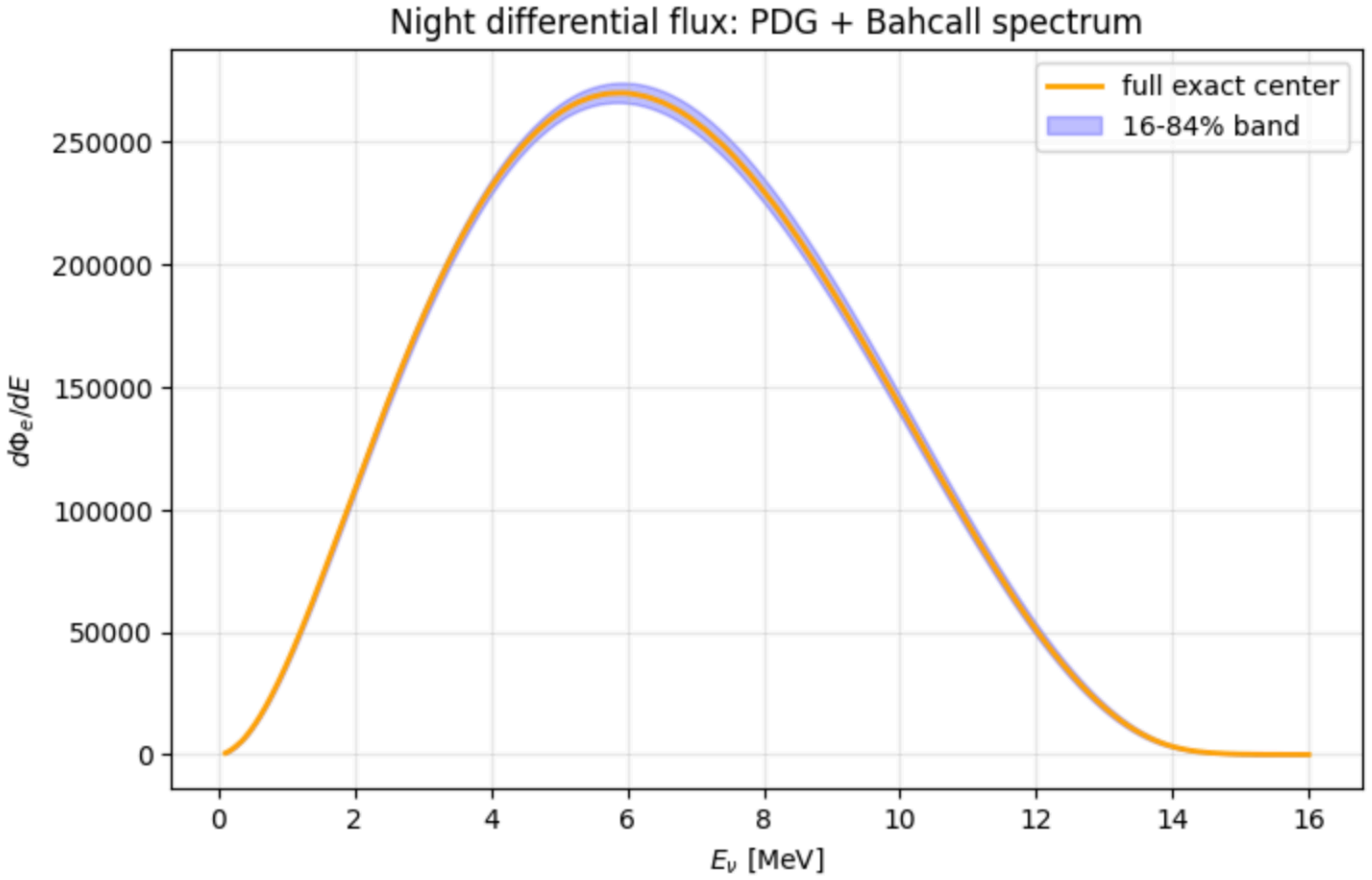
$1.81^{+0.081}_{-0.070}\,\%$, $[2,\ 16] MeV$

$2.07^{+0.092}_{-0.080}\,\%$, $[4,\ 16] MeV$

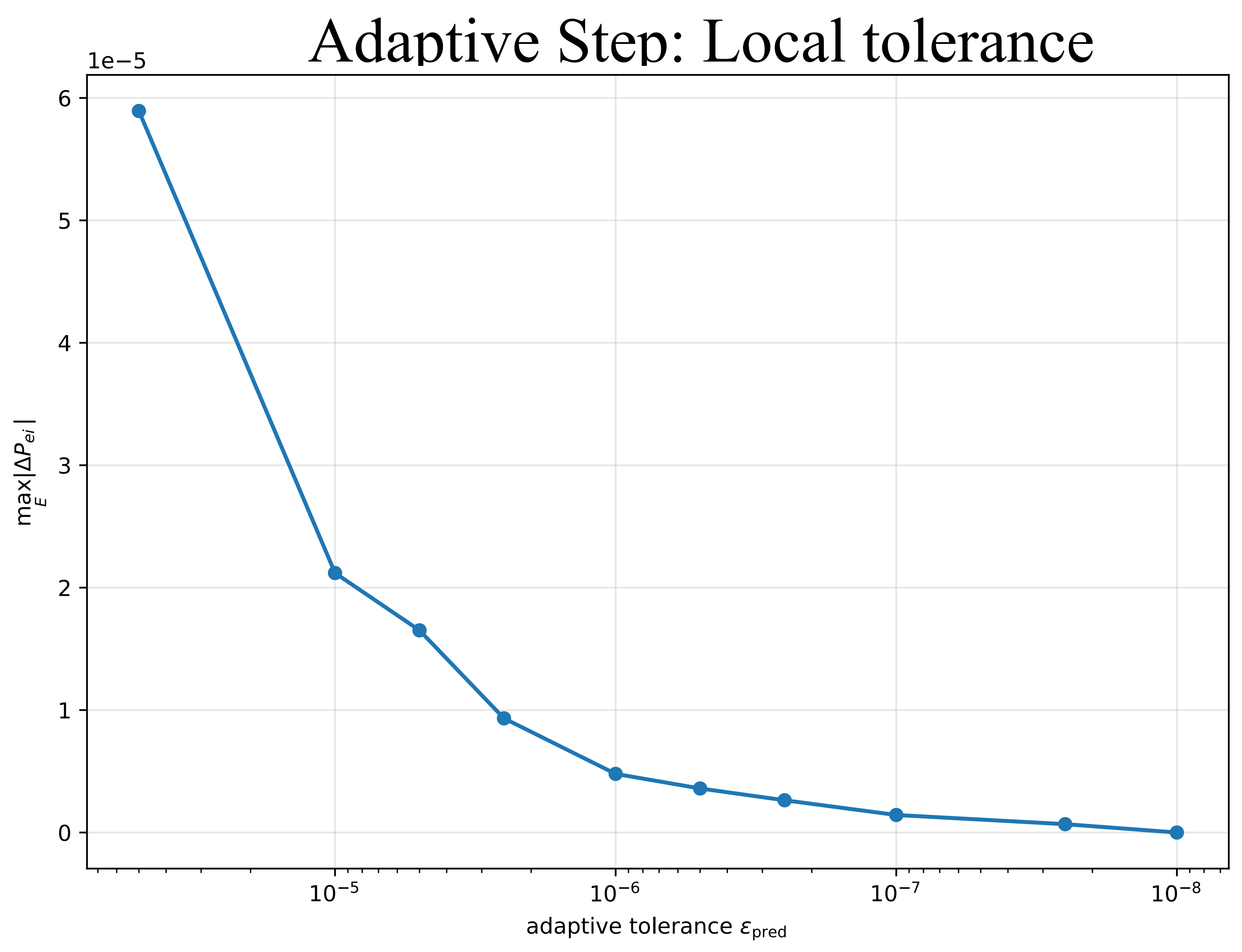
$2.28^{+0.100}_{-0.088}\,\%$, $[5,\ 16] MeV$

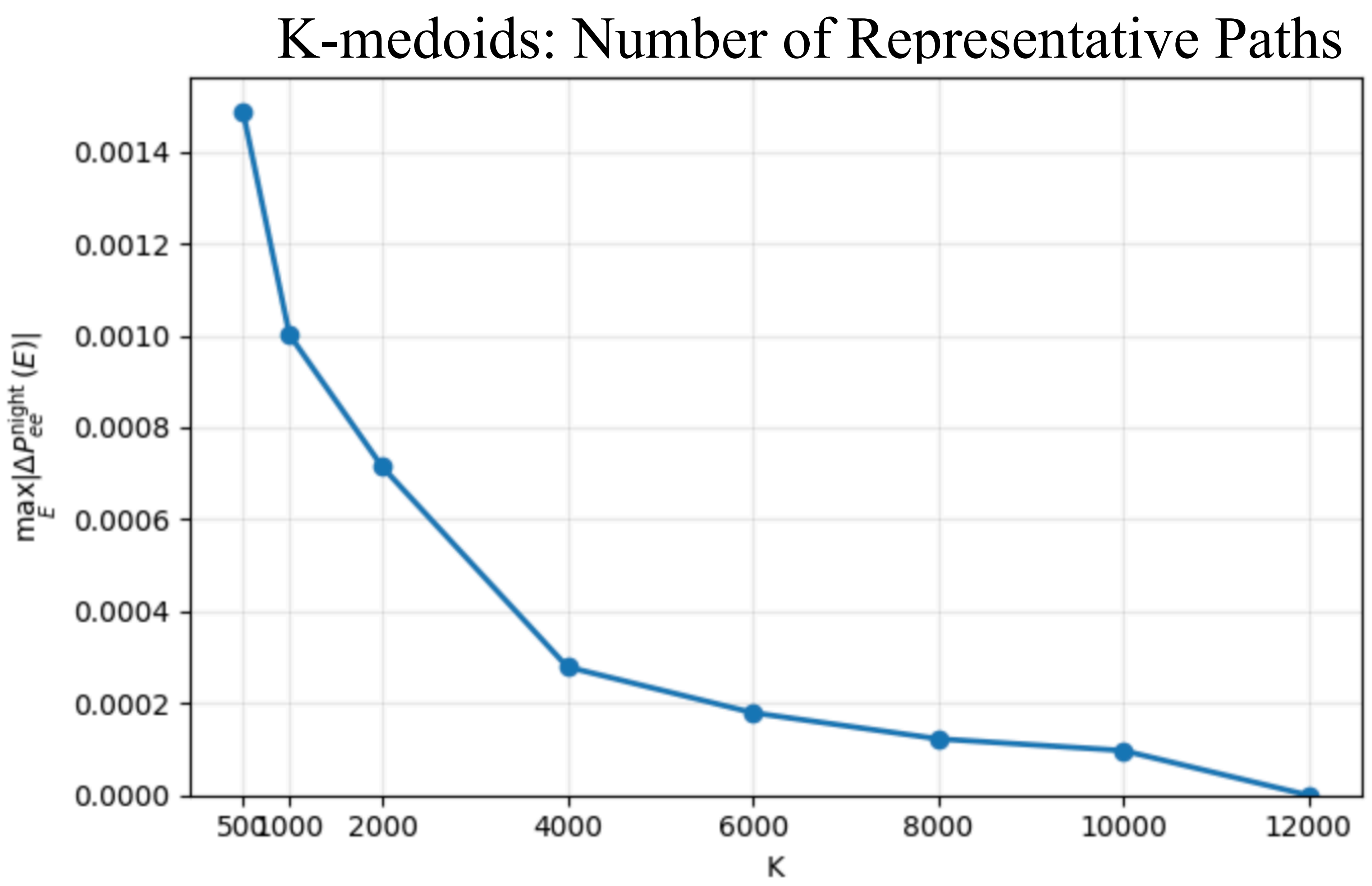
$2.53^{+0.110}_{-0.097}\,\%$, $[6,\ 16] MeV$

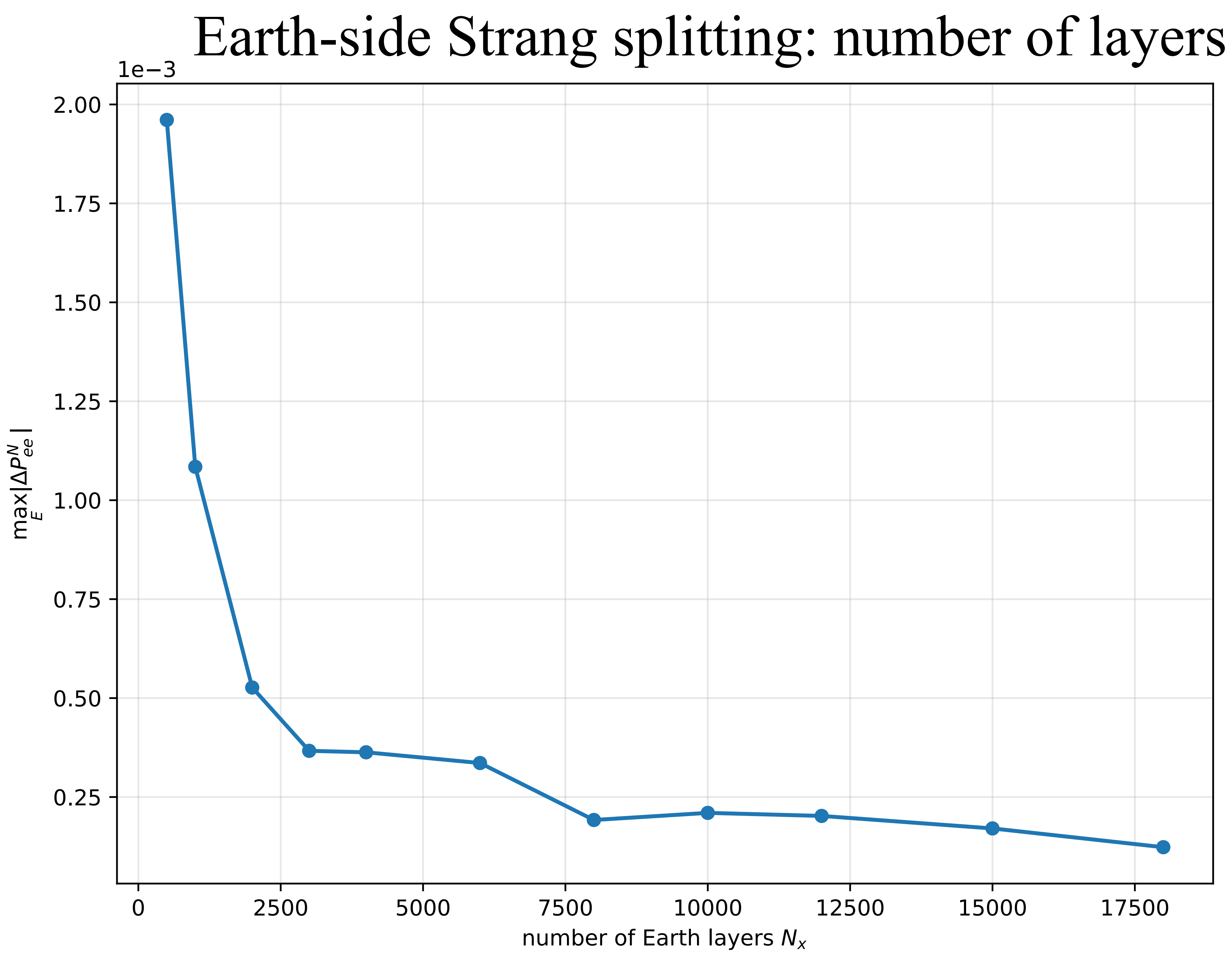




Converge Tests







Default production settings

Solar adaptive tolerance: $\varepsilon_{pred} = 5 \times 10^{-6}$

Earth Strang layers: $N_{layers}^{\oplus} = 8000$

Representative path clusters: K = 8000

Convergence is evaluated with a strict pointwise metric: the maximum absolute difference over all energy grid points for both Pee(E) and Pei(E).

Our target numerical accuracy: $\sim O(10^{-4})$

Comparison with nuSQuiDS

nuSQuIDS is used here as a general-purpose RKF reference solver.

Our Strang implementation is a specialized Earth-propagation solver:
it exploits the diagonal vacuum Hamiltonian and the rank-one matter projector in the mass basis.

Under the same hardware and single-baseline setup, the specialized Strang solver agrees with nuSQuIDS at the $\mathcal{O}(10^{-5})$ probability level and is about 35--42 $\times$ faster.

L [km]	Core crossing	t_{Strang} [s]	t_{nuSQuIDS} [s]	Speedup	$\max\lvert\Delta P\rvert$
1000	No	2.30×10^{-2}	8.00×10^{-1}	34.8×	2.36×10^{-5}
6000	No	1.39×10^{-1}	5.53	39.9×	1.22×10^{-5}
11000	Yes	2.62×10^{-1}	10.97	41.9×	3.04×10^{-5}

Summary

Our Framework provides a fast end-to-end Sun–Earth simulation framework for solar-neutrino flux predictions.

Key components:
- time-dependent Sun–Earth geometry
- finite 8B source as angular clouds
- W2-inspired k-medoids representative path clustering
- Earth-side Strang propagation with rank-one matter update
- solar-side adaptive CF4 full 3×3 propagation
- convergence-tested numerical defaults

Performance:
Single-site central prediction(3D Earth model): ~4 min 40 s (1D Earth: less than 30s)
MC uncertainty propagation: additional ~10 min

The framework makes 3D-Earth solar-neutrino predictions and uncertainty propagation computationally practical, and gives a possible workflow to calculate 3D solar density model.