Supplement to: Score-based Filtering for Land Data Assimilation

The contents of this supplement support the article "Score-based Filtering for Land Data Assimilation." It provides the model and implementation details, the pseudo-code of the sampling schemes, and the additional experimental results.

Tables

Table S1

Table S1 Moment Matching posterior score

function $\boldsymbol{s}_\theta(\boldsymbol{x}_t \mid \boldsymbol{y}, \boldsymbol{\mathcal{H}})$
$\hat{\boldsymbol{x}} \leftarrow d_\theta(\boldsymbol{x}_t, t)$
$\boldsymbol{H} \leftarrow \nabla_{\boldsymbol{x}} \boldsymbol{\mathcal{H}}(\boldsymbol{x})\big|_{\boldsymbol{x}=\hat{\boldsymbol{x}}}$
$\boldsymbol{C}_{\boldsymbol{x}|\boldsymbol{x}_t} \leftarrow \frac{1-\overline{\alpha}_t}{\sqrt{\overline{\alpha}_t}} \nabla_{\boldsymbol{x}_t} \hat{\boldsymbol{x}}$
$\boldsymbol{S}_t \leftarrow \boldsymbol{R} + \boldsymbol{H}\boldsymbol{C}_{\boldsymbol{x}|\boldsymbol{x}_t}\boldsymbol{H}^\mathsf{T}$
solve $\boldsymbol{S}_t \boldsymbol{u} = \boldsymbol{y} - \boldsymbol{\mathcal{H}}(\hat{\boldsymbol{x}})$
$\boldsymbol{s}_{\boldsymbol{y}|\boldsymbol{x}} \leftarrow (\nabla_{\boldsymbol{x}_t} \hat{\boldsymbol{x}})^\mathsf{T} \boldsymbol{H}^\mathsf{T} \boldsymbol{u}$
$\boldsymbol{s}_{\boldsymbol{x}} \leftarrow \frac{\sqrt{\overline{\alpha}_t}\hat{\boldsymbol{x}} - \boldsymbol{x}_t}{1-\overline{\alpha}_t}$
return $\boldsymbol{s}_{\boldsymbol{x}} + \boldsymbol{s}_{\boldsymbol{y}|\boldsymbol{x}}$

Table S2

Table S2 DDIM-style posterior sampling

$\boldsymbol{x}_1 \sim \mathcal{N}(\mathbf{0}, \Sigma_1)$
for $i = T$ to 1 **do**
  $s \leftarrow (i-1)/T$
  $t \leftarrow i/T$
  $\hat{\boldsymbol{x}} \leftarrow \boldsymbol{x}_t + \Sigma_t s_\theta(\boldsymbol{x}_t \mid \boldsymbol{y}, \boldsymbol{H})$ {Estimate $\mathbb{E}[\boldsymbol{x}_0 \mid \boldsymbol{x}_t, \boldsymbol{y}, \boldsymbol{H}]$}
  $\boldsymbol{z} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$
  $\boldsymbol{x}_s \leftarrow \hat{\boldsymbol{x}} + \sigma_s \sqrt{1 - \eta\big(1 - \frac{\sigma_s^2}{\sigma_t^2}\big)} \frac{\boldsymbol{x}_t - \hat{\boldsymbol{x}}}{\sigma_t} + \sigma_s \sqrt{\eta\big(1 - \frac{\sigma_s^2}{\sigma_t^2}\big)}\, \boldsymbol{z}$
end for
return $\boldsymbol{x}_0$

Table S3

Table S3: Perturbation parameter settings. Cross-correlation coefficients among the perturbed variables are listed in the last four columns. Temporal correlation is implemented via an AR(1) process.

Variable	**Perturbation type**	**Std dev**	**Timescale (h)**	**Cross correlations**			
				Prcp	**SW**	**LW**	**T**
Precipitation	Multiplicative ($e^{r-0.5\sigma^2}$)	0.5	24	1.0	−0.8	0.5	0.0
Shortwave radiation	Multiplicative ($e^{r-0.5\sigma^2}$)	0.3	24	−0.8	1.0	−0.5	0.4
Longwave radiation	Additive ($+r$)	20.0 $\mathrm{W\,m^{-2}}$	24	0.5	−0.5	1.0	0.4
Air temperature	Additive ($+r$)	1.0 K	24	0.0	0.4	0.4	1.0

Table S4

Table S4: Soil parameter perturbation settings. Here s and c denote the sand and clay percentages, respectively. The standard deviations for Ψ, Θ, and b are scaled by a factor of 0.1.

Parameter	**Texture-dependent perturbation std**	**Scaling**
Soil matric potential (Ψ)	$\log \sigma_\Psi = 0.72 - 0.0026\,[100 - (s + c)] + 0.0012\,c$	$\times 0.1$
Porosity (Θ)	$\sigma_\Theta = (7.73 - 0.073\,c)/100$	$\times 0.1$
Shape parameter (b)	$\sigma_b = 0.0500\,c + 1.34$	$\times 0.1$
Saturated hydraulic conductivity (K_{sat})	$\log \sigma_{K_{\text{sat}}} = 0.459 + 0.00321\,[100 - (s + c)]$	1.0

Table S5

Table S5: Soil moisture state perturbation settings. Perturbations are applied additively to the liquid-water content of each of the 10 soil layers ($\mathrm{kg\,m^{-2}}$). Cross-layer correlations are prescribed by a Toeplitz matrix $C_{ij} = 1 - 0.1\,|i-j|$, with covariance $\mathbf{\Sigma}_{ij} = C_{ij}\,\sigma_i\,\sigma_j$ factorized via a Cholesky decomposition. Unlike the atmospheric forcing perturbations, no temporal (AR(1)) correlation is imposed, and to conserve total water mass the residual water of the perturbed increments is transferred to the water-table storage.

Layer	**Perturbation type**	**Std dev ($\mathbf{kg\,m^{-2}}$)**
1	Additive	1.05×10^{-1}
2	Additive	1.66×10^{-1}
3	Additive	1.02×10^{-1}
4	Additive	6.61×10^{-3}
5	Additive	3.69×10^{-3}
6	Additive	1.22×10^{-3}
7	Additive	1.83×10^{-3}
8	Additive	2.61×10^{-3}
9	Additive	3.65×10^{-3}
10	Additive	4.55×10^{-3}

Figures

Figure S1

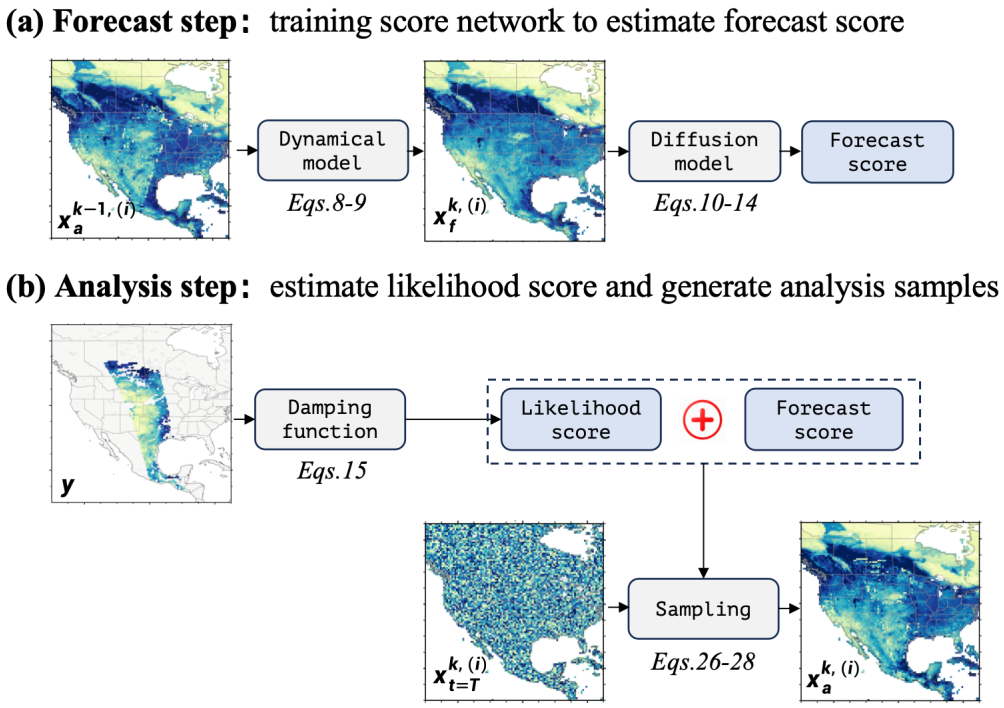

Figure S1: Schematic of the B24 forecast–analysis cycle. (a) In the forecast step, the previous analysis ensemble is propagated through the dynamical model to generate the forecast ensemble, which is used to train the diffusion model and estimate the forecast score. (b) In the analysis step, reverse diffusion is initialized from Gaussian noise, and the likelihood score is approximated by a prescribed damping function with a tunable guidance parameter. Superscripts k and i denote the assimilation time and ensemble member, respectively.

Figure S2

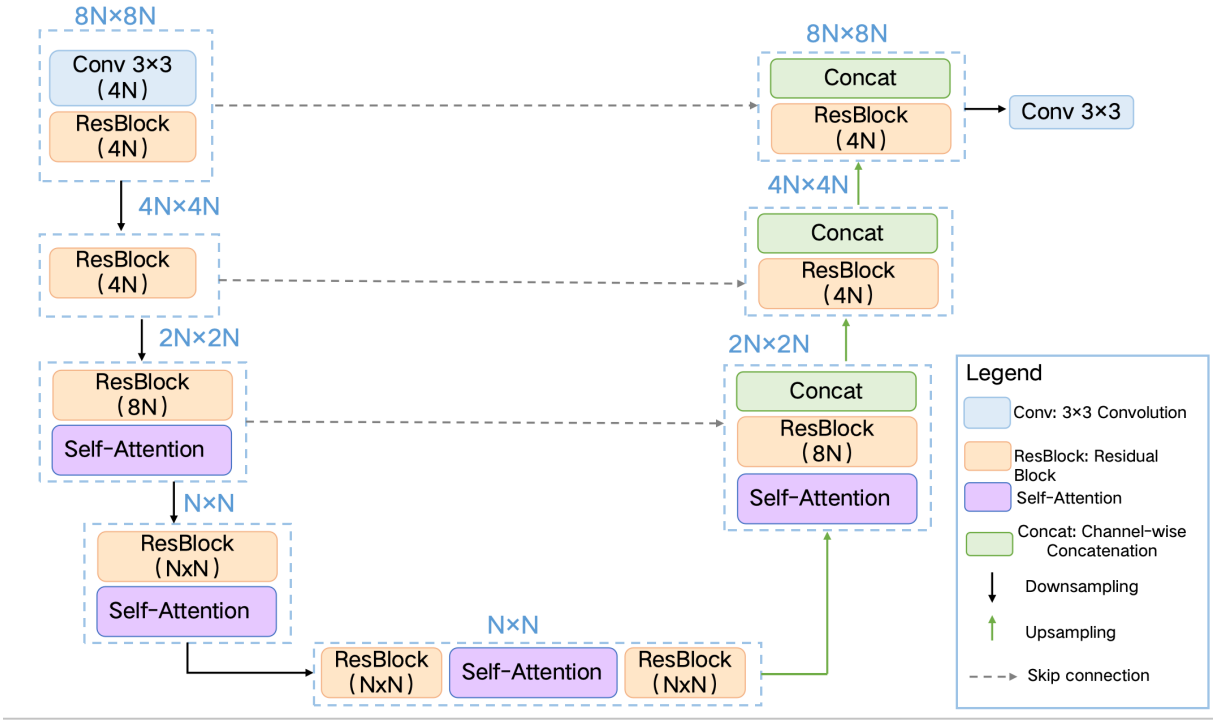

Figure S2: Architecture of the UNet-based score model used in the assimilation experiments.

Figure S3

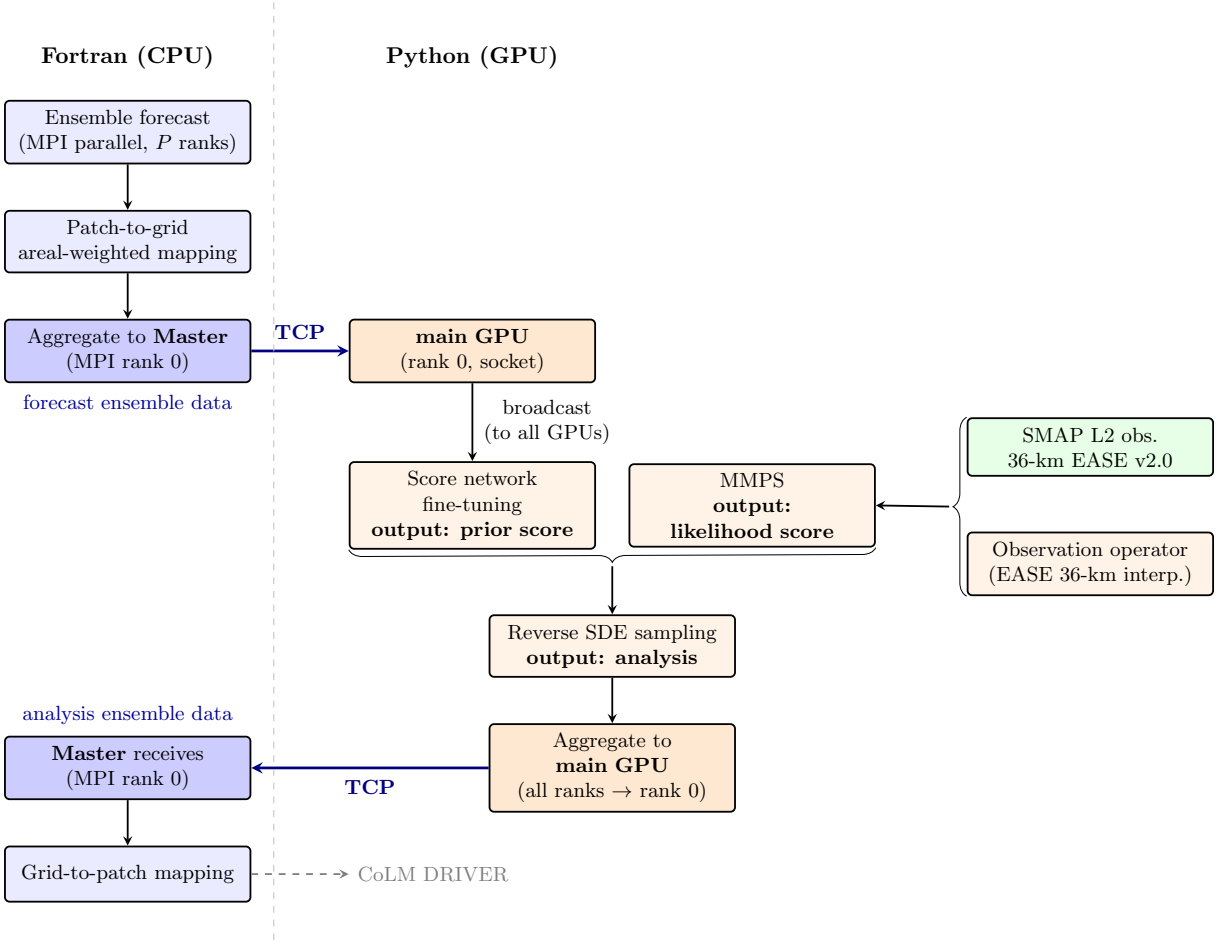

Figure S3: Schematic of the CPU-GPU coupling architecture. The left column shows the sequence of operations on the land surface model (Fortran, CPU), and the right column shows the sequence of operations in the LandSF analysis module (Python, GPU). The forecast ensemble is transferred via TCP socket to the score network, which outputs the forecast score. Meanwhile, SMAP L2 observations pass through the observation operator into MMPS, which outputs the likelihood score. The forecast and likelihood scores converge through the downward brace into reverse sampling, where they are summed to generate the analysis ensemble data, which is then returned to the land surface model via TCP socket. The dashed line indicates the next assimilation cycle.

Figure S4

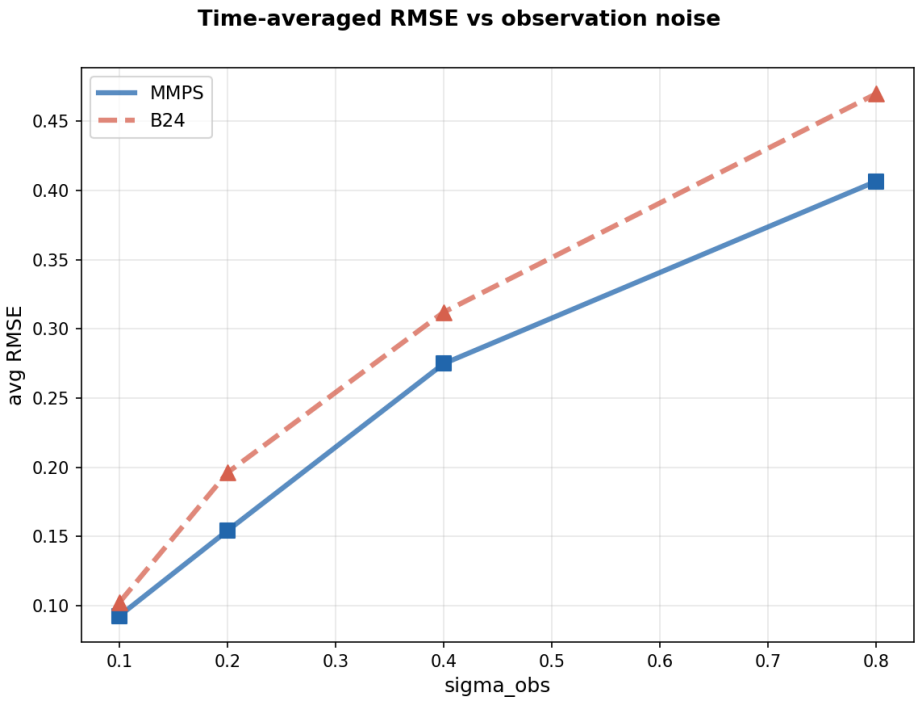

Figure S4: Time-mean spatial analysis RMSE as a function of the observation-error magnitude in the Kolmogorov-flow experiment. The RMSE is averaged over the assimilation period and over the ensemble.

Figure S5

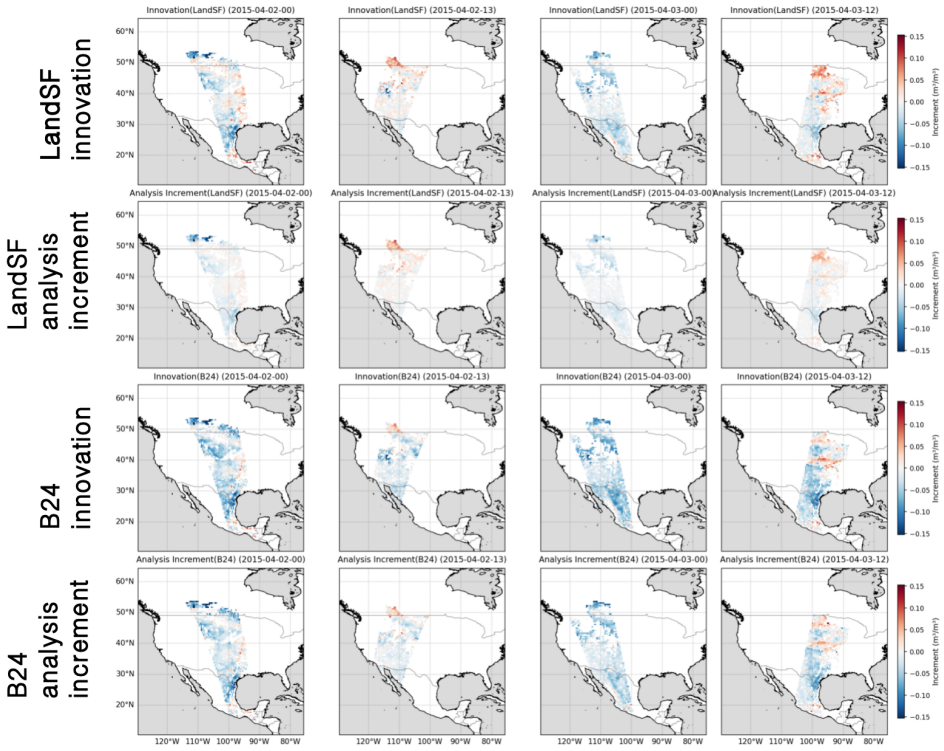

Figure S5: Spatial anomaly increments of the surface soil moisture produced by B24 and LandSF relative to the open-loop simulation.

Text S1

CPU-GPU coupling architecture

LandSF runs on heterogeneous hardware: the Fortran-based land surface model integrates the ensemble members on the CPU in parallel with MPI, while the score network fine-tuning and diffusion-based analysis sampling (Section 2.2.3) are executed on the GPU by a PyTorch-based Python implementation (Figure S3). The two sides are coupled online through a client-server protocol over a TCP socket, without intermediate files, which avoids the synchronization overhead introduced by disk I/O.

In the coupling architecture, the land surface model acts as the socket server, and the Python process of LandSF acts as the client. After the connection is established, the two sides cooperate in a request-response loop to process all assimilation times throughout the assimilation period. At each assimilation time, the model integrates the ensemble members across MPI ranks in parallel. When valid observations are detected, the liquid soil moisture in the top two layers of each member is mapped from the land surface patches to the grid via areal-weighted mapping and aggregated to the master process. The master then sends the forecast ensemble array through the socket in sequence.

On the Python side, the score network is deployed on multiple GPU processes with data parallelism, and only rank 0 holds the actual socket connection; the received data are broadcast to the remaining processes. At each assimilation time, the Python side fine-tunes the score network on the received forecast ensemble and then performs the MMPS-guided reverse diffusion sampling (Section 2.2.2) to generate N analysis members. The bidirectional interpolation between the observations (SMAP L2, 36-km EASE grid) and the model grid is performed on the GPU with tensorized operators, and grid cells outside the influence of the observations retain their forecast values. The analysis ensemble is sent back from rank 0 to the master, scattered to all ranks, and then mapped from the analysis grid back to the land surface patches. Finally, the model side converts the analysis soil moisture from volumetric water content back to mass units ($\mathrm{kg\,m^{-2}}$) and applies physical constraints such as the upper and lower bounds of porosity, so as to ensure the consistency of the model state.