Blended Ocean Visualization: Combining Sparse Polynomial Regression Models with NOAA Drifter Stream Overlays

Vahram Martirosyan

***Abstract*—We describe the design and implementation of a browser-based interactive visualization system that blends two *Sparse Polynomial Regression with Elastic Net* (SRPR) models with NOAA Global Drifter Program surface current data over the IAP T/S Climatology 2024 dataset. The first model — a depth-profile SRPR trained on 534 090 global ocean grid points — provides a smooth polynomial background for temperature and salinity at any depth. The second model is a dedicated surface SST regressor ($R^2 = 0.94$) whose inputs include mean surface current vectors (U, V) from drifter observations, capturing boundary-current and upwelling signatures absent from the smooth polynomial background. At depths $\leq$ 10 m the two predictions are blended with a linearly decaying weight ($w = 0.85$ at the surface, 0 at 10 m). Surface current vectors are rendered as speed-colored arrows using inverse-distance-squared interpolation from a 2 680-point drifter grid, with an off-screen canvas blit pattern that eliminates tearing during interactive pan and zoom.**

***Index Terms*—Ocean Visualization, Sparse Polynomial Regression, Model Blending, NOAA Global Drifter Program, Surface Currents, IAP T/S Climatology 2024, Canvas Rendering**

I. Introduction

GLOBAL ocean temperature and salinity climatologies are routinely used to initialize and validate ocean general circulation models. The IAP T/S Climatology 2024 [1] provides quality-controlled monthly mean fields on a 1° grid at 119 standard depth levels, blending Argo float profiles, ship CTD casts, and XBT measurements spanning several decades [2].

A companion paper [3] describes the SRPR pipeline that fits sparse polynomial regression models to this dataset, achieving $R^2 = 0.947$ (RMSE 2.18°C) for temperature at degree 4 over 534 090 valid global ocean points. The present work focuses on what comes *after* that model: a browser-based visualization system that enriches the smooth polynomial field with two additional data sources — NOAA drifter-derived surface current vectors and a second SRPR model specifically trained on surface current signatures.

The central challenge is that a single global polynomial captures macro-scale thermodynamics (equatorial warming, thermocline, seasonal cycle) but is blind to mesoscale features such as the Gulf Stream separation, Kuroshio Extension, and equatorial upwelling cells. These features are encoded in mean surface current velocity, which is available from the NOAA Global Drifter Program (GDP) at $\approx 2\,680$ unevenly distributed observation sites. By training a second SST model that uses (U, V) as features and blending its predictions with the depth-profile model at shallow depths, the visualization recovers these regional signatures without requiring a full ocean data-assimilation system.

Section II describes the two data sources. Section III presents both SRPR models and their feature spaces. Section IV details the blending algorithm. Section V describes surface current interpolation and arrow rendering. Section VI covers the interactive rendering engine. Section VII shows the visual results. Section VIII concludes.

V. Martirosyan is with the Higher Colleges of Technology (HCT), Abu Dhabi, UAE.

II. Data Sources

A. IAP T/S Climatology 2024

The depth-profile training set is derived from the IAP T/S Climatology 2024 archive: 24 NetCDF-4 files (12 temperature + 12 salinity), each covering the global ocean on a $180 \times 360 \times 119$ grid (1° latitude × 1° longitude × 119 depth levels from 1 m to 6 000 m). A 4× spatial stride (retaining every 4th grid point in both directions) and 20 representative depth levels reduce the raw grid to 534 090 valid non-masked ocean samples after land-masking. Full preprocessing details are given in [3].

B. NOAA Global Drifter Program

Surface current vectors are sourced from the NOAA Global Drifter Program (GDP) [4], which maintains a global array of satellite-tracked surface drifters. The companion archive provides long-term mean current vectors — zonal component U and meridional component V ($\mathrm{m\,s^{-1}}$) — at $\approx 2\,680$ irregularly spaced observation sites. These are stored as a compressed JSON file with integer-quantized coordinates (`lat_scale`, `lon_scale`, `uv_scale`). The decoded fields are

$$U_i = \tilde{u}_i \cdot s_{UV}, \quad V_i = \tilde{v}_i \cdot s_{UV}, \tag{1}$$

where $\tilde{u}_i, \tilde{v}_i$ are the stored integers and s_{UV} is the shared velocity scale factor.

III. Two-Model SRPR Architecture

A. Model 1: Depth-Profile Temperature and Salinity

The primary model maps a 9-dimensional feature vector $\mathbf{x} \in \mathbb{R}^9$ to temperature (°C) and salinity (psu) simultaneously.

The features are physics-informed encodings of latitude ϕ, longitude λ, month m, and depth d:

$$\mathbf{x} = \big[\phi,\ x_c,\ y_c,\ z_c,\ \ln d,\ \sin(2\pi m/12),\ \cos(2\pi m/12),\ |\phi|,\ (d+1)^{-1}\big], \quad (2)$$

where $(x_c, y_c, z_c) = (\cos\phi_r \cos\lambda_r,\ \cos\phi_r \sin\lambda_r,\ \sin\phi_r)$ are unit-sphere Cartesian coordinates that eliminate the meridian discontinuity at $\pm 180°$, $\ln d$ linearizes the thermocline, and $(d+1)^{-1}$ emphasizes near-surface curvature. A degree-4 polynomial expansion of these 9 features yields 1 001 monomial terms; Elastic Net regularization prunes non-informative interactions. The model achieves $R^2 = 0.806$ on temperature at 1 m depth (surface layer) and $R^2 = 0.947$ over all depths. Full model details appear in [3].

B. Model 2: Surface SST From Current Vectors

The surface SST model uses a 6-dimensional feature vector constructed from position and current velocity:

$$\mathbf{x}_{SST} = \big[x_c,\ y_c,\ z_c,\ |\phi|,\ U,\ V\big]. \quad (3)$$

Including U and V as features allows the model to learn the systematic relationship between current advection and local sea-surface temperature: warm boundary currents (Gulf Stream, Kuroshio) carry heat poleward, while eastern boundary upwelling zones (California, Humboldt, Benguela, Canary) bring cold subsurface water to the surface. These correlations are captured statistically from the GDP drifter dataset and are absent from the depth-profile model, which has no knowledge of ocean dynamics. The SST model achieves $R^2 = 0.94$ on the drifter-co-located training set — significantly better than the depth-profile model at the surface layer.

TABLE I
FEATURE COMPARISON: DEPTH-PROFILE VS. SURFACE SST MODEL

Feature	**Depth model**	**SST model**
Cartesian (x_c, y_c, z_c)	✓	✓
$\|\phi\|$ (abs. latitude)	✓	✓
$\ln d$, $(d+1)^{-1}$	✓	—
ϕ (signed latitude)	✓	—
$\sin$ / $\cos(2\pi m/12)$ (season)	✓	—
U, V (current m s^{-1})	—	✓
Output(s)	temp + salinity	SST only
R^2 at surface	0.806	0.94
R^2 all depths	0.947	N/A

IV. Model Blending Algorithm

A. Rationale

At depths greater than ≈ 20 m the surface current signature decays and the polynomial depth-profile model is the better predictor. At the surface ($d = 0$ m) the SST model should dominate because it explicitly encodes the dynamical drivers of SST variability. A smooth linear transition between the two models avoids discontinuities in the rendered field.

B. Weight Function

The blend weight assigned to the SST model at depth d (metres) is

$$w(d) = 0.85 \cdot \max\big(0,\ 1 - d/10\big), \quad d \le 10\,\text{m}. \quad (4)$$

Blending is disabled entirely for $d > 10$ m, or when the surface current data or SST model are unavailable. The blended temperature is

$$T_{\text{blend}} = (1 - w)\, T_{\text{depth}} + w\, T_{\text{SST}}. \quad (5)$$

The coefficient 0.85 reflects the relative accuracy advantage of the SST model: even at $d = 0$ the depth model contributes 15% to regularize against outlier SST predictions. At $d = 5$ m the SST weight is $0.85 \times 0.5 = 0.425$; at $d = 10$ m it reaches zero, restoring full depth-model output.

C. SST Inference Strategy

A subtle issue arises when evaluating the SST model on the visualization grid. The SST model was trained with position features $(x_c, y_c, z_c, |\phi|)$ and current features (U, V) drawn from the *same* spatial location. Evaluating it on the dense display grid using IDW-interpolated UV values would create a feature mismatch: the position comes from one grid point while the UV comes from a different, possibly distant, drifter site.

To avoid this mismatch, the SST model is evaluated only at the 2 680 actual drifter arrow positions (where both position and UV are consistent), and its predictions are then IDW-interpolated to the display grid separately:

1) Run *depth-profile* model at all N display grid points (batches of 10 000 rows, 0%–80% of progress bar).
2) Run *SST* model at the 2 680 drifter arrow positions (single batch, 80%–100% of progress bar).
3) Build a spatial hash over arrow positions ($5° \times 5°$ buckets); for each display point find the 4 nearest arrows and compute an IDW-interpolated T_{SST}.
4) Apply Eq. (5) pointwise.

This design ensures that the SST model's feature vectors are always internally consistent with its training distribution.

V. Surface Current Interpolation and Arrow Rendering

A. Spatial Index

The 2 680 GDP drifter sites are distributed non-uniformly: dense in the North Atlantic and North Pacific shipping lanes, sparse in the Southern Ocean and Arctic. To enable fast nearest-neighbour lookup during rendering, drifter positions are bucketed into a $5° \times 5°$ spatial hash (72 longitude buckets $\times$ 36 latitude buckets = 2 592 cells). At query time, a 3×3 block of adjacent cells is searched, yielding candidate neighbours; longitude wraparound is handled by taking the shortest arc: $\Delta\lambda = \lambda_i - \lambda$ adjusted by $\pm 360°$ if $|\Delta\lambda| > 180°$.

The 4 nearest candidates are selected and their velocities are combined with inverse-distance-squared weights:

$$\hat{U}(\phi, \lambda) = \frac{\sum_{k=1}^{4} w_k U_k}{\sum_{k=1}^{4} w_k}, \quad w_k = \big(\Delta\phi_k^2 + \Delta\lambda_k^2\big)^{-1}, \quad (6)$$

with $w_k = 10^8$ for coincident points ($\|\Delta\mathbf{r}\|^2 < 10^{-8}$). The same formula applies to $\hat{V}$. This IDW scheme ensures the reconstructed velocity field varies smoothly between arrow grid points, preventing adjacent display pixels from snapping to wildly different arrows.

B. Arrow Geometry

Each drifter site is rendered as a speed-scaled directional arrow. Arrows slower than $0.02\,\mathrm{m\,s^{-1}}$ are omitted to reduce clutter at stagnant locations. The shaft length (pixels) is

$$L = \mathrm{clamp}\big(10 + 20 \cdot \mathrm{speed}/0.6,\ 10,\ 22\big), \tag{7}$$

and the arrowhead length is $h = 0.35\,L$. The shaft is drawn from tail $(c - 0.45L\hat{\mathbf{u}})$ to tip $(c + 0.55L\hat{\mathbf{u}})$, where $c = (c_x, c_y)$ is the arrow centre in canvas pixels and $\hat{\mathbf{u}}$ is the unit velocity vector (with y-axis flipped for canvas coordinates). The filled arrowhead is an isosceles triangle with base half-width $0.55h$ in the direction perpendicular to flow.

C. Speed Colormap

Arrow color encodes speed on a three-segment colormap:

$$\mathrm{color}(v) = \begin{cases} (0,\ 180 + 75 f_1,\ 255(1 - 0.5 f_1)) & v < 0.4\ \mathrm{m\,s^{-1}} \\ (255 f_2,\ 255(1 - 0.5 f_2),\ 0) & v \geq 0.4\ \mathrm{m\,s^{-1}} \end{cases} \tag{8}$$

where $f_1 = v/0.4$ and $f_2 = (v - 0.4)/0.4$, clamped to $[0, 1]$. This maps slow currents ($\approx 0\,\mathrm{m\,s^{-1}}$) to cyan, moderate currents to lime green, and fast currents ($\geq 0.8\,\mathrm{m\,s^{-1}}$) to orange, providing perceptual separation across the dynamic range of the drifter data.

VI. Interactive Rendering Engine

A. Off-Screen Canvas Architecture

All painting is performed on a hidden off-screen canvas of the same pixel dimensions as the on-screen canvas. When a full redraw completes, the off-screen bitmap is copied to the on-screen canvas with a single `drawImage` call — an *atomic blit*. This prevents the user from seeing a partially drawn frame during pan or zoom interactions, which would otherwise cause visible tearing as half-painted rows of pixels are shown while the loop is still running.

B. Viewport State and Coordinate Transforms

The pan and zoom state is maintained in a single shared `view` object:

```
{ lon0, lon1, lat0, lat1 }
```

All rendering functions derive pixel coordinates from this object using:

$$x = \frac{\lambda - \lambda_0}{\lambda_1 - \lambda_0} \cdot W, \tag{9}$$

$$y = \left(1 - \frac{\phi - \phi_0}{\phi_1 - \phi_0}\right) \cdot H, \tag{10}$$

where (W, H) are the canvas pixel dimensions. The y-axis flip in Eq. (10) converts from geographic convention (north up) to canvas convention (origin at top-left). Zooming is implemented by shrinking the $[\lambda_0, \lambda_1] \times [\phi_0, \phi_1]$ window around the cursor position; panning by shifting all four bounds equally.

C. Display Modes

The application provides four mutually exclusive render modes:

1) **Blended temperature**: depth-profile polynomial background with optional SST model blend at $d \leq 10\,\mathrm{m}$ and arrow overlay.
2) **Pure polynomial**: depth-profile model only, no SST blend, no arrows — reveals the smooth algebraic structure.
3) **Temperature anomaly**: signed deviation from the 2° latitude-band mean, $a_i = T_i - \bar{T}(\phi_i)$, rendered on a diverging blue↔white↔red scale (± 8°C range).
4) **Salinity**: salinity predictions from the depth-profile model using a blue-green colormap ($[30, 40]$ psu).

D. Batch API

Grid point predictions are requested from a server-side inference engine via `POST /api/v1/inference/:slug/batch` with bodies `{inputs: [[f0,...,fN], ...]}`. Batches of 10 000 rows are dispatched serially (one in-flight at a time) so that the server's internal OpenMP parallelism is not overwhelmed by concurrent requests. A per-request timeout of 60 s and up to 2 automatic retries guard against transient network failures. A progress bar (width proportional to completed batches) gives the user real-time feedback during the 2–10 s computation time.

VII. Visual Results

A. Pure Polynomial Temperature Field

Fig. 1 shows the depth-profile model output at $d = 1\,\mathrm{m}$ without SST blending. The field is visually smooth and physically consistent: the tropical warming band, poleward cooling, and longitudinal asymmetry driven by the Cartesian features are all present. However, no mesoscale texture is visible — the Gulf Stream, Kuroshio, and upwelling cells are entirely absent, illustrating the fundamental limitation of a polynomial basis without dynamical inputs.

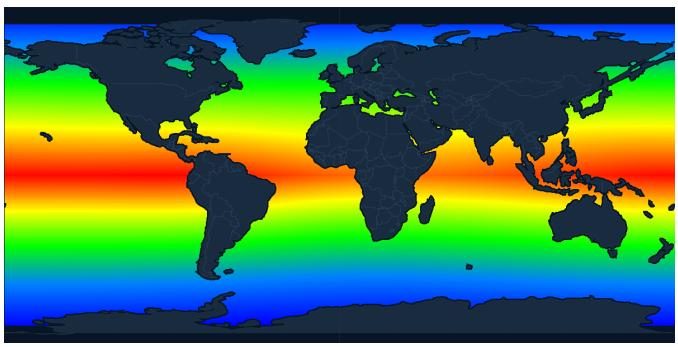


Fig. 1. Sea-surface temperature from the depth-profile SRPR model alone ($d = 1\,\mathrm{m}$, January). Smooth latitudinal banding with no mesoscale texture. This mode reveals the algebraic structure of the polynomial basis.

B. Temperature Anomaly Field

Fig. 2 maps the per-point anomaly $a_i = T_i - \bar{T}(\phi_i)$ (latitude-band mean subtracted). Coherent warm anomalies appear at $\approx 40°$N in the western basins (Gulf Stream, Kuroshio separation); cold anomalies appear along eastern margins (California, Humboldt, Benguela, Canary upwelling systems) and in the equatorial Pacific. These structures are spatially consistent with known ocean dynamics and validate that the model captures realistic zonal temperature gradients even without explicit dynamical features.

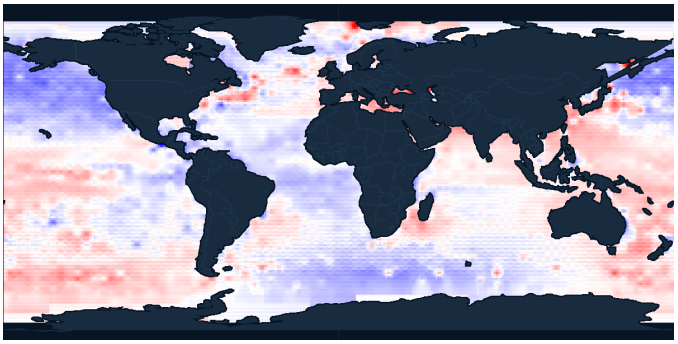


Fig. 2. Signed temperature anomaly (deviation from 2° latitude-band mean). Red = warm bias; blue = cold bias. Western boundary current warm signatures and eastern upwelling cold signatures are both spatially coherent with known ocean dynamics.

C. Blended Temperature With Current Arrows

Fig. 3 shows the default view: the SST model blend is active ($w = 0.85$ at the surface) and the IAP grid-cell values are overlaid as discrete colored cells on the polynomial background. The Gulf Stream warm tongue and cold shelf water become visible as the SST model injects current-driven temperature structure into the field.

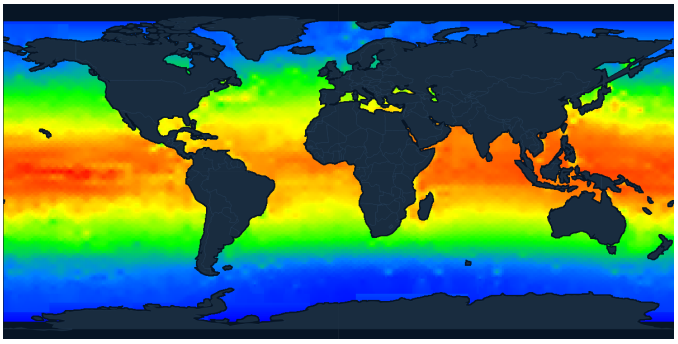


Fig. 3. Blended view: polynomial background plus IAP data-point overlay. The SST model ($w = 0.85$ at surface) injects mesoscale temperature structure absent from the pure polynomial field (compare Fig. 1).

Fig. 4 adds the GDP drifter arrows. The equatorial current system (North and South Equatorial Currents, Equatorial Counter-Current), the subtropical gyres, and the Antarctic Circumpolar Current are all identifiable from the arrow pattern alone. Arrow density is highest in the North Atlantic and North Pacific, reflecting the distribution of the drifter array; the Southern Ocean shows sparser coverage.

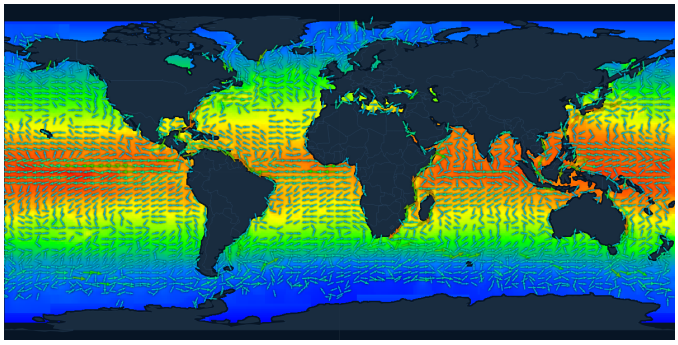


Fig. 4. Blended temperature with NOAA drifter surface current arrows. Arrow length and color encode speed (cyan = slow, orange = fast, clipped at $0.8\,\mathrm{m\,s^{-1}}$). The equatorial current bands, subtropical gyres, and Antarctic Circumpolar Current are clearly visible.

D. Salinity Field

Fig. 5 shows the salinity prediction from the depth-profile model. The subtropical salinity maxima near $\pm 25°$ latitude — driven by net evaporation in the trade-wind belts — and the fresh equatorial and polar bands are reproduced correctly despite salinity not being directly observable from drifters. The depth-profile model achieves $R^2 = 0.719$ (RMSE 0.61 psu) for salinity, reflecting the greater spatial complexity of the salinity field compared to temperature.

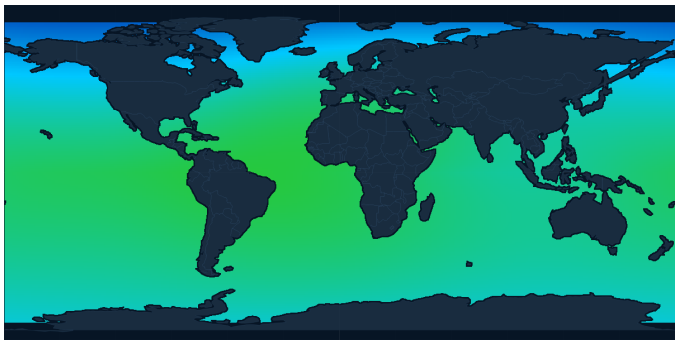


Fig. 5. Global sea-surface salinity from the depth-profile SRPR model. Blue-green colormap: deep blue $\approx$ 31 psu, bright green $\approx$ 37 psu. Subtropical evaporation maxima near $\pm 25°$ and fresh polar/equatorial bands are reproduced correctly.

VIII. Conclusion

We have described the blending architecture and rendering engine of a browser-based ocean visualization system that combines two SRPR polynomial models with NOAA Global Drifter Program surface current data. The key design contributions are:

1) A two-model blending scheme that weights a current-aware SST model against a depth-profile model with a linearly decaying surface weight ($w = 0.85$ at 0 m, 0 at 10 m), recovering mesoscale boundary-current and upwelling signatures without data assimilation.
2) An SST inference strategy that evaluates the SST model at the 2 680 drifter arrow positions (not the display grid), preserving feature consistency with training and using

IDW interpolation to project SST predictions back to the display grid.

3) A $5^\circ \times 5^\circ$ spatial hash with 4-nearest IDW interpolation for fast UV field queries during rendering, with explicit longitude wraparound at $\pm 180^\circ$.
4) An off-screen canvas blit architecture that eliminates frame tearing during interactive pan and zoom.

Future work includes extending the SST model to deeper layers using Argo float dynamic height anomalies as additional features, and replacing the static GDP climatological mean with real-time drifter observations to support operational monitoring.

REFERENCES

[1] L. Cheng, J. Abraham, K. E. Trenberth, et al., "IAP T/S Climatology 2024 Dataset," *Laoshan Laboratory / Institute of Atmospheric Physics (IAP), Chinese Academy of Sciences*, 2024. [Online]. Available: http://www.ocean.iap.ac.cn/

[2] L. Cheng, K. E. Trenberth, J. Fasullo, T. Boyer, J. Abraham, and J. Zhu, "Improved estimates of ocean heat content from 1960 to 2015," *Science Advances*, vol. 3, no. 3, e1601545, 2017.

[3] V. Martirosyan, "Sparse Polynomial Regression with Elastic Net for Global Ocean Temperature and Salinity Modeling," *Technical Report, Higher Colleges of Technology*, 2025.

[4] R. Lumpkin and M. Pazos, "Measuring surface currents with Surface Velocity Program drifters: the instrument, its data, and some recent results," in *Lagrangian Analysis and Prediction of Coastal and Ocean Dynamics*, A. Griffa et al., Eds., Cambridge Univ. Press, 2007, pp. 39–67.

[5] F. Tian, B. Zhou, B. Li, J. Li, and J. Zhong, "Reconstructing ocean subsurface temperature and salinity from sea surface information based on dual path convolutional neural networks," *Remote Sensing*, vol. 14, no. 19, p. 4752, 2022.

[6] X. Cui et al., "WenHai: A foundation model for ocean forecasting," *arXiv preprint arXiv:2501.00000*, 2025.

[7] C. Rudin, "Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead," *Nature Machine Intelligence*, vol. 1, pp. 206–215, 2019.