Transport-Informed Gradient Fields for DREAMPlace

Bhrugu Bharathi, Tanish Talapaneni, Nithin Baimeedi

May 2025

1 Motivation

After experiencing difficulty reproducing the architecture developed in "Chip Placement with Diffusion Models" [9], we returned to the description of Project 2 to reconsider our approach. On reread, we were drawn to the language describing a density-induced pushing force as one of the optimization drivers: "Most state-of-the-art placers [...] formulate the placement problem as a nonlinear optimization problem and iteratively optimize cell locations using a wirelength-induced pulling force and a density-induced pushing force." This prompted us to consider whether the electrostatic density-induced pushing paradigm could be improved by revisiting another physical force analog.

We opted to investigate a fluid-like analog, as the RePlace placement animations resembled a fluid flow across the die area. Initially, we considered modeling cell movement using the Navier–Stokes equations, under the intuition that macros and cells could be treated as a viscous fluid. However, Navier–Stokes introduces significant complexity: it requires solving for both pressure and velocity fields and demands strict stability conditions for numerical simulation. Moreover, chip placement lacks intrinsic viscosity and turbulent behavior. We concluded that a formulation that described mass movement guided by global regularity superseded fine-scale flow features.

2 Problem Formulation

While exploring alternative fluid models, we encountered the 2-Wasserstein distance W_2, specifically in its dynamic formulation Benamou-Brenier (BB). Unlike static metrics, BB provides a time-continuous transport plan between mass distributions by minimizing the total kinetic energy of motion [1][2].

The Benamou–Brenier formulation is obtained from the Monge–Kantorovich [7][8] W_2 formulation subject to the following constraints:

Let $\rho(t, x)$ denote the time-varying **mass density function** over the placement domain $\Omega \subset \mathbb{R}^2$, where $t \in [0, 1]$ and $x \in \Omega$. Formally, we define:

$$\rho : [0,1] \times \Omega \to \mathbb{R}_{\geq 0}$$

such that $\rho(t,x)$ gives the amount of mass (cell or macro density) at position x at time t.

Let $\mathbf{v}(t,x)$ be the **Eulerian velocity field**, representing the instantaneous velocity of mass at location x and time t:

$$\mathbf{v} : [0,1] \times \Omega \to \mathbb{R}^2$$

Let us also introduce the **Lagrangian flow map** $\Phi(t,x)$, which tracks the trajectory of a mass particle that starts at position x at time $t = 0$:

$$\Phi : [0,1] \times \Omega \to \Omega$$

with the condition that:

$$\frac{d}{dt}\Phi(t,x) = \mathbf{v}(t, \Phi(t,x))$$

That is, particles follow the velocity field $\mathbf{v}$ through Ω over time.

These quantities must satisfy the **continuity equation**, which enforces the conservation of mass throughout the domain:

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{v}) = 0 \quad \text{on } [0,1] \times \Omega$$

In other words, the rate at which the mass density changes at a given location must equal the net outflow of mass from that point. The second quantity is described by the divergence of the mass flux.

Given initial and target mass distributions $\rho_0(x)$ and $\rho_1(x)$ - corresponding to the current and target placements, respectively - the Benamou-Brenier formulation solves the following dynamic optimization problem:

$$\inf_{\rho, \mathbf{v}} \left\{ \int_0^1 \int_\Omega \rho(t,x) \|\mathbf{v}(t,x)\|^2 \, dx \, dt \right\}$$

subject to:

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{v}) = 0$$

$$\rho(0,x) = \rho_0(x), \quad \rho(1,x) = \rho_1(x)$$

This formulation defines a continuous-time transport plan that displaces the mass from ρ_0 to ρ_1 in a way that minimizes the total kinetic energy expended.

Solving this formulation directly yields an optimal velocity field $\mathbf{v}(t,x)$ and a density path $\rho(t,x)$ over time - in other words, the complete geodesic between ρ_0 and ρ_1 in the Wasserstein space. Obtaining this solution requires solving a constrained optimization problem using Lagrange multipliers.

In our case, we are not interested in recovering the full transport plan over time. Instead, we aim to extract a single, physically meaningful velocity field that represents how mass should move, instantaneously, from its current configuration ρ_0 toward a more uniform target ρ_1. This velocity field will be used to compute gradients in the DREAMPlace optimization loop [5][6], effectively replacing the electrostatic force with one derived from an optimal transport-inspired principle.

As such, we do not need to solve the entire Benamou–Brenier program. Instead, we seek an approximation of the initial-time velocity field $\mathbf{v}(0, x)$ that best reflects the direction of steepest descent in transport cost from ρ_0 to ρ_1.

To do so, we revisit the continuity equation:

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{v}) = 0$$

At time $t = 0$, we approximate the time derivative $\frac{\partial \rho}{\partial t}$ using a finite difference approximation: we treat the time derivative as the instantaneous difference between the target density and the current density:

$$\frac{\partial \rho}{\partial t} \approx \rho_1(x) - \rho_0(x)$$

Substituting into the continuity equation gives:

$$\rho_1(x) - \rho_0(x) + \nabla \cdot (\rho_0(x)\mathbf{v}(x)) = 0$$

This can be rearranged as a weighted Poisson equation for the velocity potential $\phi(x)$, assuming $\mathbf{v}(x) = -\nabla\phi(x)$. Here, $\phi(x)$ plays the role of a scalar potential, and the velocity field is recovered as its negative gradient: $\mathbf{v}(x) = -\nabla\phi(x)$:

$$\nabla \cdot (\rho_0(x)\nabla\phi(x)) = \rho_0(x) - \rho_1(x)$$

This formulation resembles the electrostatic Poisson equation used in DREAMPlace, but we posit that it respects mass conservation and produces a velocity field derived from optimal transport considerations.

Solving this weighted Poisson equation yields a gradient field $\mathbf{v}(x) = -\nabla\phi(x)$, which we can then use to push cells in the direction of transport-informed descent.

3 Multigrid Solution Method

Having established the weighted Poisson formulation:

$$\nabla \cdot (\rho_0(x)\nabla\phi(x)) = \rho_0(x) - \rho_1(x)$$

we now address solving this equation efficiently on large chip placement domains. Direct methods become prohibitively expensive for realistic problem

sizes, while simple iterative methods converge slowly due to the multiscale nature of placement problems—from fine-grained standard cells to large macros.

We use geometric multigrid [3][4], which exploits how different error frequencies converge at different rates. High-frequency errors are eliminated by local smoothing operations, while low-frequency errors require global information transfer through coarse grid correction.

3.1 Discretization

We discretize the weighted Poisson equation on a uniform grid with spacing h_x and h_y. The operator $\nabla \cdot (\rho_0 \nabla \phi)$ uses conservative finite differences with harmonic averaging of density weights at cell faces.

For interior grid point (i, j), the discrete operator becomes:

$$\frac{1}{h_x^2}\left[\rho_e(\phi_{i+1,j} - \phi_{i,j}) - \rho_w(\phi_{i,j} - \phi_{i-1,j})\right] + \frac{1}{h_y^2}\left[\rho_n(\phi_{i,j+1} - \phi_{i,j}) - \rho_s(\phi_{i,j} - \phi_{i,j-1})\right]$$

where the face-centered densities use harmonic averaging:

$$\rho_e = \frac{2\rho_{i,j}\rho_{i+1,j}}{\rho_{i,j} + \rho_{i+1,j}}, \quad \rho_w = \frac{2\rho_{i,j}\rho_{i-1,j}}{\rho_{i,j} + \rho_{i-1,j}}$$

and similarly for ρ_n and ρ_s. This harmonic averaging handles discontinuous density fields that arise from macro boundaries and empty regions.

3.2 V-Cycle Algorithm

The multigrid method constructs a hierarchy of coarser grids, each with spacing doubled in both dimensions. On each level, we perform:

Pre-smoothing: Apply Red-Black Gauss-Seidel with successive over-relaxation (SOR) to reduce high-frequency error:

$$\phi_{i,j}^{new} = \phi_{i,j}^{old} + \omega\left(\frac{\text{neighbor contributions} + f_{i,j}}{\text{diagonal coefficient}} - \phi_{i,j}^{old}\right)$$

where $\omega \approx 1.2$ is the over-relaxation parameter.

Residual computation: Calculate $r = f - L\phi$ where L is the discrete weighted Laplacian.

Restriction: Transfer the residual to the coarser grid using full-weighting with a 9-point stencil:

$$\frac{1}{16}\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

Coarse grid solve: Recursively apply V-cycle, or solve directly for grids $\leq 4 \times 4$.

Prolongation: Interpolate the coarse correction back using bilinear interpolation.

Post-smoothing: Additional smoothing to eliminate prolongation artifacts.

3.3 Boundary Conditions

We use Dirichlet boundary conditions ($\phi = 0$) at domain boundaries, ensuring the transport velocity $v = -\nabla\phi$ has zero normal component at boundaries, preventing mass flow outside the chip area.

4 Implementation Optimizations

4.1 Vectorized Operations

We implement smoothing using PyTorch's vectorized operations with TorchScript compilation. The key insight is replacing nested loops with `torch.roll` operations for neighbor access:

```
# Get neighbor values using circular rolls
phi_east = torch.roll(phi, -1, dim=0)
phi_west = torch.roll(phi, 1, dim=0)
phi_north = torch.roll(phi, -1, dim=1)
phi_south = torch.roll(phi, 1, dim=1)
```

This leverages GPU parallelism and provides significant speedup over traditional implementations.

4.2 Red-Black Ordering

The Gauss-Seidel smoother uses red-black ordering, updating grid points in a checkerboard pattern for parallelization and improved convergence:

```
red_mask = (i + j) % 2 == 0
black_mask = ~red_mask
```

4.3 Adaptive Grid Hierarchy

The multigrid hierarchy automatically adapts to the size of the problem, coarsening until 4×4 grids suitable for direct solution.

5 Integration with DREAMPlace

5.1 Gradient Field Computation

Once solved for ϕ, the transport velocity field uses central finite differences:

$$v_x = -\frac{\partial \phi}{\partial x} \approx -\frac{\phi_{i+1,j} - \phi_{i-1,j}}{2h_x}$$

$$v_y = -\frac{\partial \phi}{\partial y} \approx -\frac{\phi_{i,j+1} - \phi_{i,j-1}}{2h_y}$$

This velocity field gives the optimal instantaneous direction for cell movement to reduce density violations while conserving mass.

5.2 Force Application

The velocity field $v(x)$ replaces the electrostatic density force in DREAMPlace. For each cell at position x_c:

$$F_{\text{transport}} = \alpha \cdot v(x_c)$$

where α balances the transport force against wirelength and other objectives.

5.3 Computational Complexity

The multigrid solver achieves $O(N)$ complexity for N grid points, compared to $O(N^{3/2})$ for direct methods. Convergence typically requires 1-10 V-cycles, but remains computationally expensive, requiring up to 20 seconds on moderately sized grids (4096×4096) even with minimal smoothing.

References

[1] Jean-David Benamou and Yann Brenier. A numerical method for the optimal time-continuous mass transport problem and related problems. *Contemp. Math.*, 226:1–11, 1998.

[2] Jean-David Benamou and Yann Brenier. A computational fluid mechanics solution to the Monge-Kantorovich mass transfer problem. *Numerische Mathematik*, 84(3):375–393, 2000.

[3] William L. Briggs, Van Emden Henson, and Steve F. McCormick. *A Multigrid Tutorial*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2nd edition, 2000.

[4] Ulrich Trottenberg, Cornelius W. Oosterlee, and Anton Schuller. *Multigrid*. Academic Press, London, 2000.

[5] Yibo Lin, Shounak Dhar, Wuxi Li, Haoxing Ren, Brucek Khailany, and David Z. Pan. DREAMPLACE: Deep learning toolkit-enabled GPU acceleration for modern VLSI placement. In *Proceedings of the 56th Annual Design Automation Conference* (DAC '19), pages 1–6. ACM, 2019.

[6] Yibo Lin, Wuxi Li, Jiaqi Gu, Haoxing Ren, Brucek Khailany, and David Z. Pan. ABCDPlace: Accelerated batch-based concurrent detailed placement on multi-threaded CPUs and GPUs. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 39(5):936–948, 2020.

[7] Gaspard Monge. Mémoire sur la théorie des déblais et des remblais. *Histoire de l'Académie Royale des Sciences de Paris*, pages 666–704, 1781.

[8] Leonid V. Kantorovich. On the translocation of masses. *Comptes Rendus (Doklady) de l'Académie des Sciences de l'URSS*, 37:199–201, 1942.

[9] Vint Lee, Minh Nguyen, Leena Elzeiny, and Chun Deng. Chip placement with diffusion models. Master's thesis, EECS Department, University of California, Berkeley, 2025.