Geometric Flow Models over Neural Network Weights

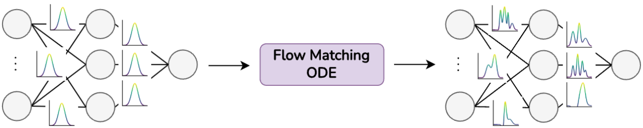
Ege Erdogan

February 6, 2025

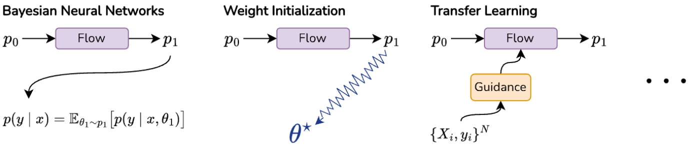
TUM MSc Thesis

Large amount of work on weight-space generative models, e.g. (Wang et al., 2025).

Existing work overlooks the geometry of NN weights, or only models permutation symmetries.



We build fully geometric generative models accounting for both permutation and scaling symmetries.



Flow Matching (Lipman et al., 2023)

Goal: Learn a time-dependent **vector field** $v_\theta : [0,1] \times \mathbb{R}^d \to \mathbb{R}^d$ that transports density p_0 to p_1.

Design Choices & Training

Given samples $x_0 \sim p_0, x_1 \sim p_1$, define:

1. **Coupling** $q(x_0, x_1)$: $p(x_0)p(x_1)$
2. **Probability path** $p_t(x_t \mid x_0, x_1)$: $\mathcal{N}(x_t \mid (1-t)x_0 + tx_1, \sigma^2)$
3. "True" **vector field** $u_t(x_t \mid x_0, x_1)$: $x_1 - x_0$

Optimize the **conditional flow matching (CFM)** objective:

$$\theta^* = \arg\min_{\Theta} \mathbb{E}_{t\sim\mathcal{U}[0,1],(x_0,x_1)\sim q, x_t\sim p_t} \left[\| v_\theta(t, x_t) - u_t(x_t \mid x_0, x_1) \|^2 \right]$$

Sampling

Integrate the ODE $dx = v_\theta(t, x_t)dt$.

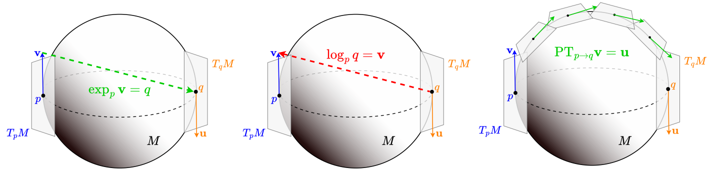
Optimal Transport Couplings (Tong et al., 2023)

$$q(x_0, x_1) := \pi(x_0, x_1) \quad \pi := \text{ approx. OT map}$$

Flow model then approximates the dynamic OT map from p_0 to p_1. Can lead to straighter/shorter trajectories.

Riemannain Flow Matching (Chen and Lipman, 2023) Model the vector field over Riemannian manifolds.

$$x_t := \exp_{x_0}(t \log_{x_0} x_1) \quad u_t(x_t \mid x_0, x_1) := \frac{\log_{x_t} x_1}{1 - t}$$



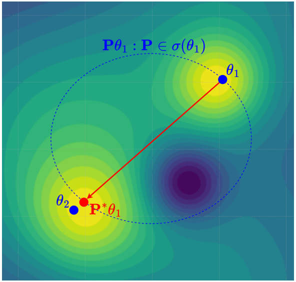
Permutation symmetries between architectural components.

e.g. between subsequent layers in MLPs.

Scaling symmetries from non-linear activations.

e.g. ReLU: $ReLU(\lambda x) = \lambda ReLU(x) \quad \lambda \geq 0$

Linear Mode Connectivity Hypothesis: Low-loss solutions linearly connected up to permutation symmetries.

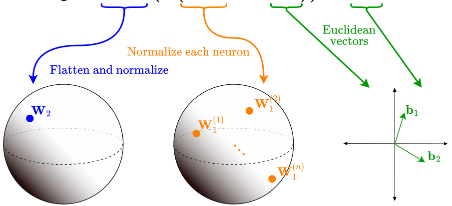


For ReLU MLPs:

1. **Align** all NNs to a single reference NN via rebasin.
2. **Normalize** incoming weights of each neuron, and inversely multiply the outgoing weights.

Both operations preserve the function the NN computes.

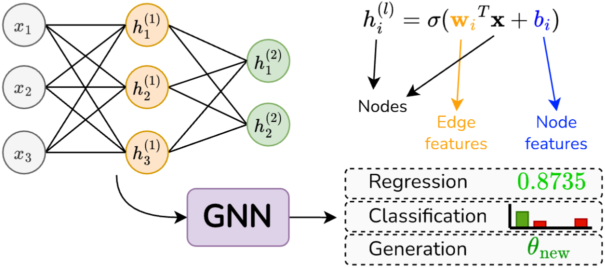
$$\hat{y} = \mathbf{W_2}\,(\sigma(\mathbf{W_1}\mathbf{x} + \mathbf{b_1})) + \mathbf{b_2}$$



$\implies$ Neurons on the hypersphere.
$\implies$ Last layer on the hypersphere.
$\implies$ Biases as Euclidean vectors.

Neural networks can be modeled as **graphs** through their computational graphs.

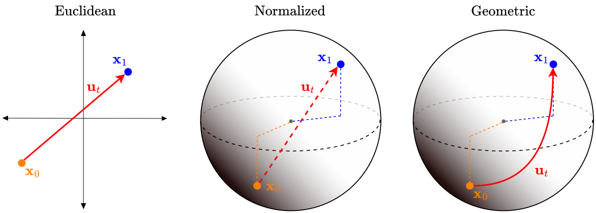
Can be processed using **graph neural networks**.



We use the Relational Transformer (with edge updates) (Kofinas et al., 2024; Diao and Loynd, 2023)

Flows in Weight-Space

Pre-processing: Align all weights to a reference (rebasin).



Euclidean. Use the weights w/o further processing.

Normalized. Normalization + vector field in Euclidean space (i.e. inside the hyperpsheres).

Geometric. Normalization + vector field on the product geometry (Riemannain flow matching).

Training

p_0: Zero-mean Gaussian. p_1: Sampled from SGD trajectories.

Couplings: Independent, mini-batch OT.

Sample $t \sim \text{Beta}(1,2)$ rather than uniformly, to optimize early time points (higher loss) for more steps.

Sampling

Integrate ODEs with Euler solver:

$$x_0 \sim p_0 \quad , \quad x_{t+\Delta t} = x_t + v_\theta(x_t, t)\Delta t$$

Optional guidance with gradients from the base task:

$$x_{t+\Delta t} = x_t + \left(v_\theta(x_t, t) + \lambda \nabla_{x_t} \mathcal{L}(f, x_t)\right) \Delta t$$

Results

One-Shot Performance on Easier Tasks

Two-hidden-layer MLP (30-16-16-2) on the UCI Wisconsin Breast Cancer dataset, binary classification.

Flow	**Accuracy**	**Loss**
Euclidean	0.998 ± 0.006	0.101 ± 0.050
Euclidean (aligned)	0.998 ± 0.006	0.070 ± 0.040
Euclidean (aligned + OT)	0.993 ± 0.010	0.053 ± 0.028
Normalized	0.993 ± 0.009	0.027 ± 0.014
Normalized (aligned)	0.989 ± 0.011	0.030 ± 0.015
Normalized (aligned + OT)	0.988 ± 0.018	0.044 ± 0.047
Geometric	0.992 ± 0.011	0.019 ± 0.009
Geometric (aligned)	0.993 ± 0.001	0.018 ± 0.001
Geometric (aligned + OT)	0.991 ± 0.011	0.020 ± 0.012
Target	0.992 ± 0.010	0.048 ± 0.032

All flows can sample high-quality individual weights, matching or exceeding Adam-optimized weights.

MNIST - Sample Quality and Diversity

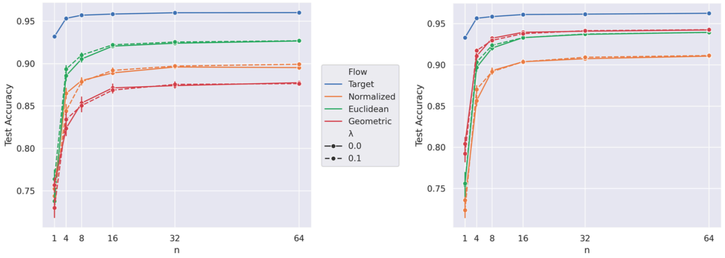
Larger MLP (784-10-10) on MNIST, 10 classes.

Trained with $\sim 60K$ samples. 512 Euler steps to sample.

Flow	**Accuracy**	**Loss**
Euclidean (aligned)	0.737 ± 0.085	0.814 ± 0.286
Euclidean (aligned + OT)	0.757 ± 0.077	0.753 ± 0.245
Normalized (aligned)	0.753 ± 0.074	1.537 ± 0.046
Normalized (aligned + OT)	0.706 ± 0.078	1.608 ± 0.047
Geometric (aligned)	0.737 ± 0.070	1.457 ± 0.040
Geometric (aligned + OT)	0.786 ± 0.064	1.443 ± 0.046
Target	0.933 ± 0.009	0.231 ± 0.027

- Generated weights less accurate than optimized weights.
- Geometric flow with OT couplings performs the best.

Average predictions over generated weights:



(a) Independent Couplings (b) OT Couplings

- Significantly more accurate than individual weights.
- OT couplings improve performance in all setups.
- Guidance has little effect.

Transferability - Datasets

Use the MNIST flow for Fashion-MNIST (same architecture).
Three approaches:

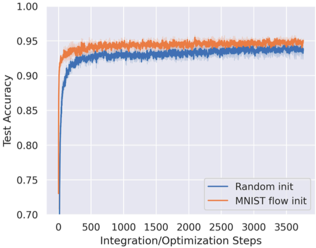
1. Use the generated weights directly.
2. Guide sampling with gradients from Fashion-MNIST.
3. Init model with generated weights and train on Fashion-MNIST.

Guide sampling with gradients from Fashion-MNIST:

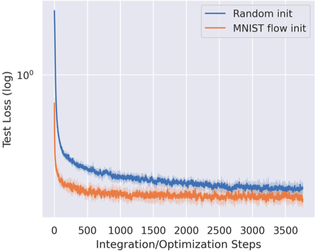
Flow	Accuracy	Loss
Adam-optimized	0.908 ± 0.003	0.334 ± 0.008
With Guidance		
Euclidean	0.754 ± 0.082	0.764 ± 0.252
Normalized	0.724 ± 0.081	1.543 ± 0.045
Geometric	0.730 ± 0.067	1.470 ± 0.043
No guidance	0.080 ± 0.030	9.601 ± 1.402

- Generated weights themselves perform poorly as expected.
- Guidance for 512 steps significantly helps. Generated weights reach the same accuracies they do on MNIST now on a task harder to learn.

Initialize model using weights generated with the MNIST flow (w/o guidance). Then train on Fashion-MNIST:



(a) Test Accuracy



(b) Test Loss

- Training converges faster, although towards similar levels of performance.

A GNN is not limited to certain graph structures.

Sample weights for a 784-32-10 MLP using the flow trained on 784-10-10 weights, both on MNIST.

Flow	Accuracy	Loss
Euclidean	0.826 ± 0.076	0.830 ± 0.281
Euclidean (w/ guidance)	0.842 ± 0.079	0.796 ± 0.337
Geometric	0.890 ± 0.030	1.030 ± 0.038
Geometric (w/ guidance)	0.886 ± 0.033	1.031 ± 0.037
Normalized	0.203 ± 0.122	2.652 ± 0.667
Normalized (w/ guidance)	0.184 ± 0.083	2.776 ± 0.575

- Normalized flow fails but Euclidean and Geometric flows succeed.
- **Generated weights perform better than the weights the flow was trained on.**

Contributions

Geometry of NN weights can be utilized to build generative models.

Such models can generalize to different tasks and architectures.

Future Directions

- **Model further symmetries**. Different architectures, activations, data-dependent symmetries...
- **Flows over distributions** (e.g. Meta Flow Matching (Atanackovic et al., 2024)) $\rightarrow$ weight-space "foundation models"?
- **Guidance beyond task gradients**. Further differentiable objectives, condition on desired losses, ...
- **Training without samples**, given the likelihood function (work in this line: (Akhound-Sadegh et al., 2024)).

Thesis & Slides: `erdogan.dev/thesis.pdf` `/thesis_slides.pdf`

Akhound-Sadegh, Tara et al. (2024). *Iterated Denoising Energy Matching for Sampling from Boltzmann Densities*. DOI: 10.48550/arXiv.2402.06121. arXiv: 2402.06121 [cs, stat]. URL: http://arxiv.org/abs/2402.06121. Pre-published.

Atanackovic, Lazar et al. (2024). *Meta Flow Matching: Integrating Vector Fields on the Wasserstein Manifold*. DOI: 10.48550/arXiv.2408.14608. arXiv: 2408.14608 [cs, stat]. URL: http://arxiv.org/abs/2408.14608. Pre-published.

- Chen, Ricky T. Q. and Yaron Lipman (2023). *Riemannian Flow Matching on General Geometries*. DOI: 10.48550/arXiv.2302.03660. arXiv: 2302.03660 [cs, stat]. URL: http://arxiv.org/abs/2302.03660. Pre-published.
- Diao, Cameron and Ricky Loynd (2023). *Relational Attention: Generalizing Transformers for Graph-Structured Tasks*. DOI: 10.48550/arXiv.2210.05062. arXiv: 2210.05062 [cs]. URL: http://arxiv.org/abs/2210.05062. Pre-published.

Kofinas, Miltiadis et al. (2024). *Graph Neural Networks for Learning Equivariant Representations of Neural Networks*. DOI: 10.48550/arXiv.2403.12143. arXiv: 2403.12143 [cs, stat]. URL: http://arxiv.org/abs/2403.12143. Pre-published.

Lipman, Yaron et al. (2023). *Flow Matching for Generative Modeling*. DOI: 10.48550/arXiv.2210.02747. arXiv: 2210.02747 [cs, stat]. URL: http://arxiv.org/abs/2210.02747. Pre-published.

Pittorino, Fabrizio et al. (2022). "Deep Networks on Toroids: Removing Symmetries Reveals the Structure of Flat Regions in the Landscape Geometry". In: *Proceedings of the 39th International Conference on Machine Learning*. International Conference on Machine Learning. PMLR, pp. 17759–17781.

Tong, Alexander et al. (2023). *Improving and Generalizing Flow-Based Generative Models with Minibatch Optimal Transport*. DOI: 10.48550/arXiv.2302.00482. arXiv: 2302.00482 [cs]. URL: http://arxiv.org/abs/2302.00482. Pre-published.

Wang, Kai et al. (2025). “Recurrent Diffusion for Large-Scale Parameter Generation”. In: *arXiv preprint arXiv:2501.11587*.