The Convex Connection

Bridging Deep Learning and Optimal Transport

Mouad ID SOUGOU & Omar ARBI
Generative Modeling
MVA

March 26, 2025

Overview

Introduction

- Convex functions are extensively studied due to their properties, particularly their gradients.
- Present in domains like linear inverse problems and optimal transport
- Unstead of manually designing them, deep learning can be a great tool in order to learn convex functions and then their gradients

Two architectures that learn monotone gradients of convex functions, without first learning the underlying convex function or its Hessian. [Chaudhari et al., 2023]

Learning a Monotone Gradient Function

We want to learn a monotone gradient function $g(x) : \mathbb{R}^n \to \mathbb{R}^n$ such that

$$g(x) = \nabla f(x)$$

for some convex, twice differentiable function $f(x) : \mathbb{R}^n \to \mathbb{R}$.
A function $f(x)$ is convex if and only if its Hessian is positive semi-definite (PSD):

$$H_f(x) = J_g(x) \succeq 0, \quad \forall x \in \mathbb{R}^n.$$

Therefore, when parameterizing $g(x)$ using a neural network, its Jacobian must be PSD with respect to the input to guarantee convexity.

Section 2

Proposed architectures

Cascaded Monotone Gradient Network (C-MGN)

The first proposed architecture is a Cascaded Monotone Gradient Network (C-MGN) formulated as follows:

C-MGN Formulation

$$z_0 = Wx + b_0 \tag{1}$$
$$z_\ell = Wx + \sigma_\ell(z_{\ell-1}) + b_\ell \tag{2}$$
$$\texttt{C-MGN}(x) = W^\top \sigma_L(z_{L-1}) + V^\top Vx + b_L \tag{3}$$

where the layer outputs z_ℓ, biases b_ℓ, and activation functions σ_ℓ may vary across layers ℓ, but **all L layers share the weight matrix W**.

Modular Monotone Gradient Network (M-MGN)

The second architecture introduced is the Modular Monotone Gradient Network (M-MGN), formulated as follows:

M-MGN Formulation

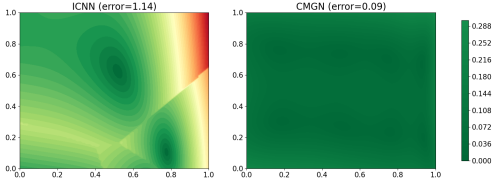
$$z_k = \mathbf{W}_k x + b_k \tag{4}$$

$$\texttt{M-MGN}(x) = a + V^\top V x + \sum_{k=1}^{K} s_k(z_k)\mathbf{W}_k^\top \sigma_k(z_k) \tag{5}$$

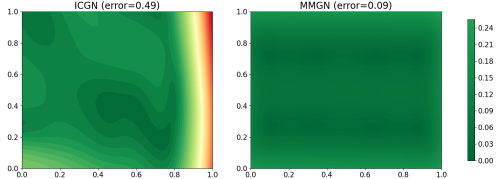
If the scalar-value function s_k is **convex, twice differentiable, and nonnegative** and can be expressed as $\sigma_k(\cdot) = \nabla s_k(\cdot)$, then the Jacobian of the M-MGN with respect to its input is PSD.

Experiments in 2D dimension

Estimate the gradient field of a classical function using 10000 data points sampled from the unit square, 50 epochs and with an Adam Optimizer with 0.001 as learning rate.



(a) C-MGN and I-CGN



(b) M-MGN and I-CNN

Figure: l_2 error maps between the learned gradient and the true one for the four architectures.

	C-MGN	M-MGN	I-CGN	I-CNN
Number of Parameters	14	24	15	76
RMSE	0.09	0.09	0.49	1.14

$\Rightarrow$ C-MGN and M-MGN learn better and with **fewer parameters**

Section 3

Optimal Transport

Optimal Transport Problem

- Find a transport plan T that pushes a probability measure α onto another one β, while minimizing the total transport cost c

$$\inf_T \int_{\mathcal{X}} c(x, T(x)) d\alpha(x), \quad T_{\#}\alpha = \beta. \tag{6}$$

- Important result is in the case of euclidian loss, where the transport is defined as the gradient of a convex function

Theorem (Brenier)

Let $\mathcal{X} = \mathcal{Y} = \mathbb{R}^d$ and consider the quadratic cost function $c(x, y) = \|x - y\|^2$, and under specific conditions, there exists a unique optimal transport map $T : \mathbb{R}^d \to \mathbb{R}^d$. Moreover, this map is the gradient of a convex function φ, meaning $T(x) = \nabla\varphi(x)$.

Experiments in Gaussian setup

$\mathbf{x}_{\text{source}} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ and the target one is $\mathcal{N}(\mathbf{0}, \mathbf{I}_2)$ where:

$$\boldsymbol{\mu} = \begin{bmatrix} 2 \\ 1 \end{bmatrix}, \quad \boldsymbol{\Sigma} = \begin{bmatrix} 1 & -0.8 \\ -0.8 & 1 \end{bmatrix}.$$

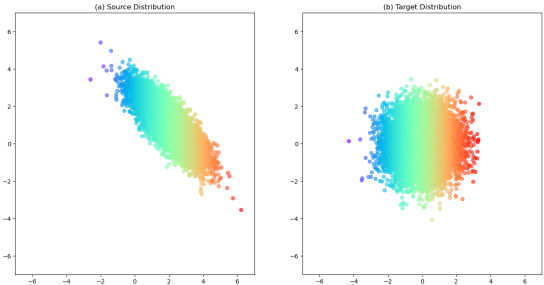
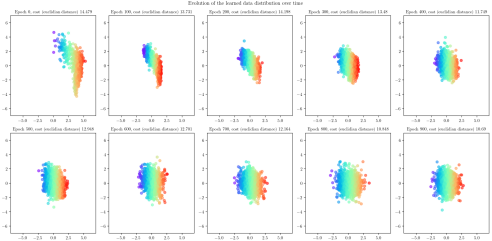


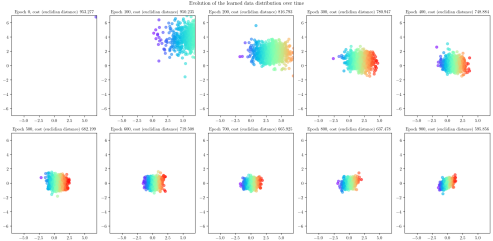
Figure: Source and Target Distribution

Experiments in Gaussian setup

We used both the C-MGN and M-MGN to learn the mapping g and compare it with the the theoretical one.



(a) C-MGN



(b) M-MGN

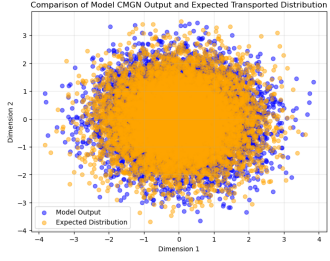
Figure: Evolution of the learned data distribution over time

$\Rightarrow$ Both architectures learn the mapping, but M-MGN do it with a higher cost

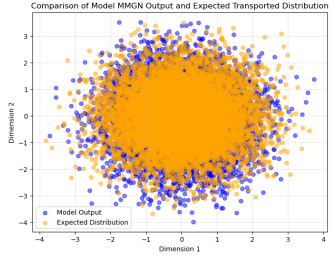
Experiments in Gaussian setup

The **closed-form optimal transport map** for Gaussian distribution corresponds to an affine map [Peyré and Cuturi, 2019] :

$$g^*(x) = \Sigma_X^{-1/2}(x - \mu_X).$$



(a) C-MGN vs. Closed-Form Solution



(b) M-MGN vs. Closed-Form Solution

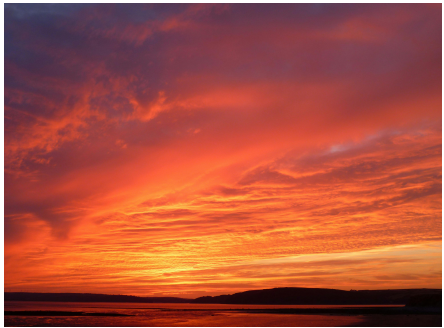
$\Rightarrow$ The learned transport maps $g(x)$ closely match the theoretical optimal map $g^*(x)$

Subsection 3

Color Domain Adaptation

Dataset

(a) Input Image: Dark Zurich Dataset



(b) Sunset Image: Target Color Distribution

Figure: Model Inputs for Optimal Transport

Target Distribution

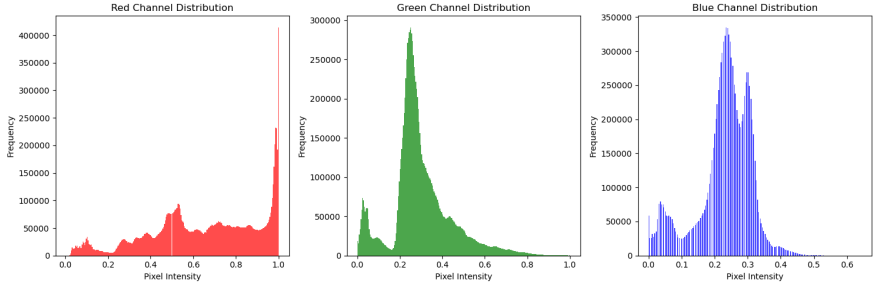


Figure: RGB Color Distribution for the Sunset Image

The distribution is not inherently a multivariate Gaussian; however, for the sake of simplicity, we will approximate it as one in our initial analysis.

M-MGN for Color Transport

Trained an M-MGN model with 28 parameters on a low-budget CPU (learning rate: 0.01) for 50 epochs, achieving effective color transport early in training and strong results upon completion.

Figure: M-MGN Output for the Dark Zurich Input Image at Epoch 50

(a) Transported Colors at Epoch 1

(b) Transported Colors at Epoch 50

Figure: Comparison of Model Outputs at Different Training Stages

$\Rightarrow$ M-MGN demonstrates effective color transport early in training.

This outcome is linked to the consistently low training loss observed from the start.

To assess the model's generalization, we evaluate it on an unseen image from the Dark Zurich dataset.

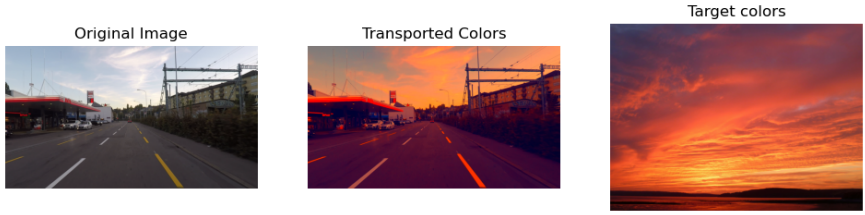


Figure: M-MGN Model Output for an Unseen Image from the Dark Zurich Dataset

$\Rightarrow$ The results suggest that the learned mapping is not overfitted to the training data and can be applied to other images within the same domain.

C-MGN for Color Transport

Trained a C-MGN model with 27 parameters over 100 epochs (lr $= 0.01$). Two runs were conducted with different initializations for matrix V.
For context, the role of matrix V is described below:

C-MGN Formulation

$$\texttt{C-MGN}(x) = W^\top \sigma_L(z_{L-1}) + V^\top V x + b_L \tag{7}$$

(a) Transported Colors at first Epoch with Unconstrained Initialization of V

(b) Transported Colors at first Epoch with Orthogonal Initialization of V

Training Metrics Across the Two Runs

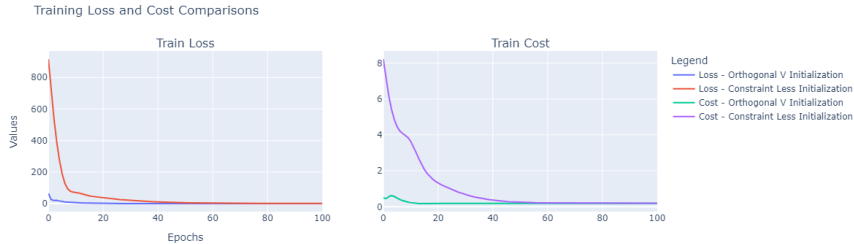


Figure: Training Loss and Cost for C-MGN under Different V Initializations

$\Rightarrow$ Orthogonal initialization shows better stability and convergence, achieving lower training loss and cost. Both initializations ultimately produce similar results, comparable to those of M-MGN.

After training we test our model on an unseen image from the Dark Zurich Dataset:

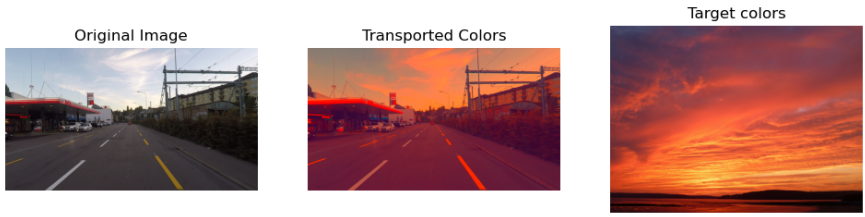


Figure: C-MGN Model Output for an Unseen Image from the Dark Zurich Dataset

$\Rightarrow$ The results suggest that the model generalizes well to unseen images from the same domain.

Modeling Target Distribution as a GMM

Target distribution is now modeled as a mixture of multivariate Gaussians to capture complex patterns.

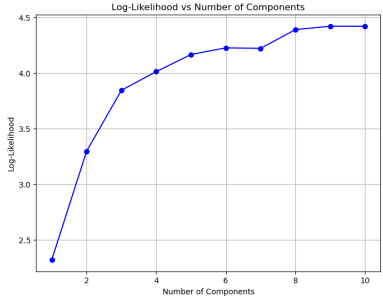


Figure: Log-Likelihood vs. Number of Components for the Sunset Image Target Distribution

Using the Elbow method, we determined 5 components as an optimal balance between model complexity and log-likelihood.

M-MGN Training with Gaussian Mixture Target

Trained an M-MGN model with 27 parameters for 50 epochs using the Adam optimizer ($\text{lr} = 0.01$) and Kullback-Leibler Divergence as the loss function. The final training loss of 24.8 is higher than for a single Gaussian target, reflecting the added complexity of the Gaussian mixture.

Figure: M-MGN Output for a Gaussian Mixture Target Distribution

M-MGN: Nighttime Transformation with Northern Lights

(a) Input: Snowy mountains with clear blue skies.

(b) : Nighttime scenes with northern lights.

Setup:

- Target distribution: Multivariate Gaussian.
- Training: Same parameters and optimization as prior experiments.
- Loss function: KL divergence.

Results

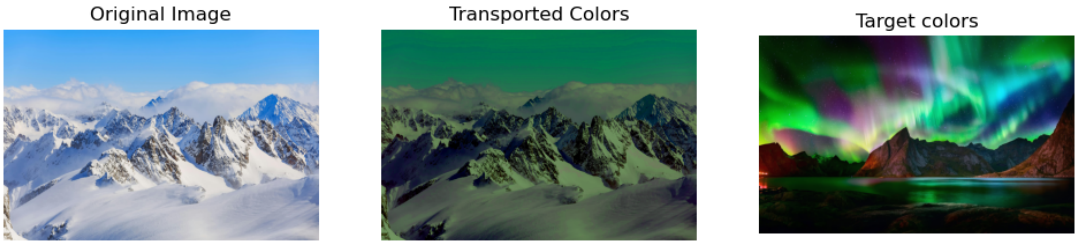


Figure: Color transport results using M-MGN.

M-MGN achieves a train loss of 0.18, successfully mapping daytime to nighttime scenes.
⇒ However, structural bright areas persist, suggesting future refinements in texture adaptation and spatial coherence.

Conclusion

Key Findings:

- Explored Monotone Gradient Networks (MGNs) for learning convex gradients.
- Examined C-MGN and M-MGN architectures, showcasing their efficacy in preserving monotonicity and representing convex functions.
- Demonstrated MGN's application in optimal transport tasks, successfully transforming daytime mountain images to nighttime scenes with northern lights.

Insights and Future Work:

- Results highlight M-MGN's ability to capture global color transformations while indicating areas for improvement in texture adaptation and structural consistency.
- Potential future directions include refining MGNs for high-dimensional tasks and extending their application to complex optimal transport problems.

The End

References


Chaudhari, S., Pranav, S., and Moura, J. M. (2023).
Learning gradients of convex functions with monotone gradient networks.
Electrical and Computer Engineering.


Peyré, G. and Cuturi, M. (2019).
Computational Optimal Transport, volume 11 of *Foundations and Trends in Machine Learning*.
Now Publishers.