DAGMA: Learning DAGs via M-matrices and a Log-Determinant Acyclicity Characterization

Kevin Bello

Bryon Aragam

Pradeep Ravikumar

THE UNIVERSITY OF CHICAGO

ML MACHINE LEARNING DEPARTMENT Carnegie Mellon University

NEURAL INFORMATION PROCESSING SYSTEMS

Problem introduction

Problem introduction

- The goal is to learn the underlying **directed acyclic graph (DAG)** of a structural equation model (SEM). A *Markovian* nonparametric SEM consists of a set of equations of the form,

$$X_j = f_j(X, Z_j),\ \forall j \in [d],$$

where each $f_j : \mathbb{R}^{d+1} \rightarrow \mathbb{R}$ is a nonparametric function, and Z_j represents noise.

Problem introduction

- The goal is to learn the underlying **directed acyclic graph (DAG)** of a structural equation model (SEM). A *Markovian* nonparametric SEM consists of a set of equations of the form,

$$X_j = f_j(X, Z_j),\ \forall j \in [d],$$

  where each $f_j : \mathbb{R}^{d+1} \to \mathbb{R}$ is a nonparametric function, and Z_j represents noise.

- E.g., linear SEMs: $X_j = w_j^\top X + Z_j$, where $W = [w_1 \mid \cdots \mid w_d]$ represents the weighted adjacency matrix.

Problem introduction

- The goal is to learn the underlying **directed acyclic graph (DAG)** of a structural equation model (SEM). A *Markovian* nonparametric SEM consists of a set of equations of the form,

$$X_j = f_j(X, Z_j),\ \forall j \in [d],$$

where each $f_j : \mathbb{R}^{d+1} \to \mathbb{R}$ is a nonparametric function, and Z_j represents noise.

- E.g., linear SEMs: $X_j = w_j^\top X + Z_j$, where $W = [w_1 \mid \cdots \mid w_d]$ represents the weighted adjacency matrix.

$$\begin{array}{cccc} X_1 & X_2 & X_3 & X_4 \end{array}$$
$$\begin{bmatrix} 1.00 & -2.14 & 0.87 & -1.82 \\ -1.5 & 0.39 & 0.45 & -0.09 \\ \vdots & \vdots & \vdots & \vdots \end{bmatrix}$$

Problem introduction

- The goal is to learn the underlying **directed acyclic graph (DAG)** of a structural equation model (SEM). A *Markovian* nonparametric SEM consists of a set of equations of the form,

$$X_j = f_j(X, Z_j),\ \forall j \in [d],$$

where each $f_j : \mathbb{R}^{d+1} \to \mathbb{R}$ is a nonparametric function, and Z_j represents noise.

- E.g., linear SEMs: $X_j = w_j^\top X + Z_j$, where $W = [w_1 \mid \cdots \mid w_d]$ represents the weighted adjacency matrix.

$$\begin{array}{cccc} X_1 & X_2 & X_3 & X_4 \end{array}$$
$$\begin{bmatrix} 1.00 & -2.14 & 0.87 & -1.82 \\ -1.5 & 0.39 & 0.45 & -0.09 \\ \vdots & \vdots & \vdots & \vdots \end{bmatrix}$$

estimate →

Problem introduction

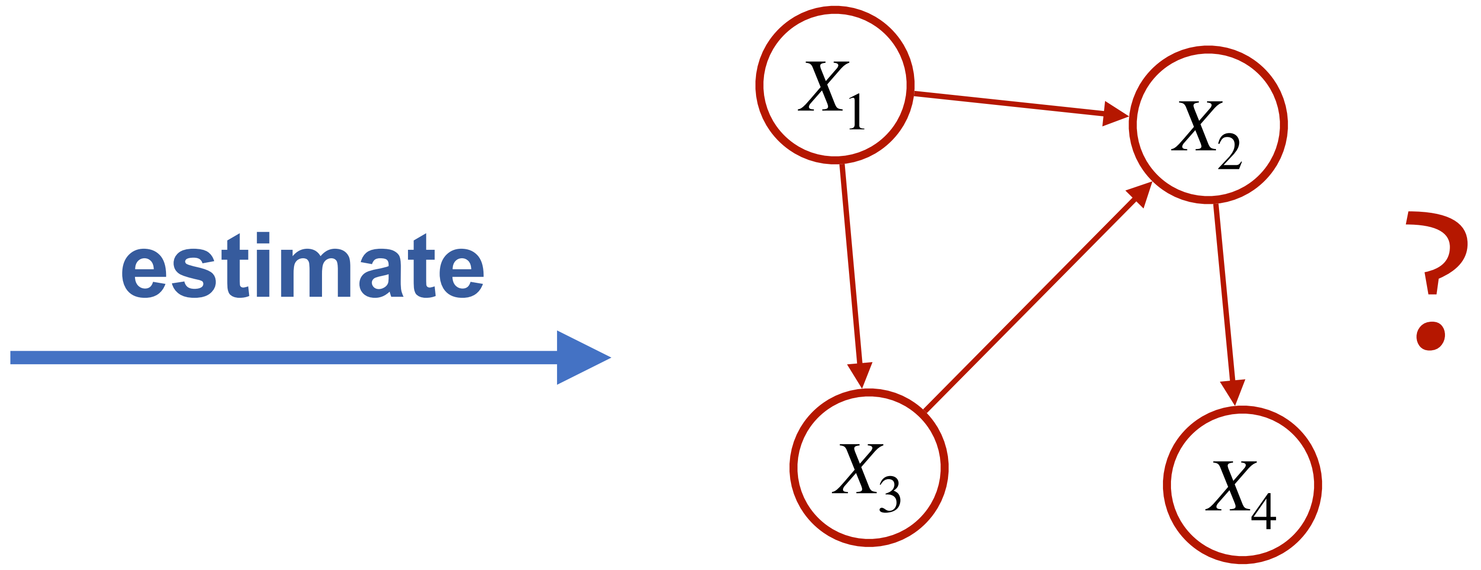
- The goal is to learn the underlying **directed acyclic graph (DAG)** of a structural equation model (SEM). A *Markovian* nonparametric SEM consists of a set of equations of the form,

$$X_j = f_j(X, Z_j),\ \forall j \in [d],$$

where each $f_j : \mathbb{R}^{d+1} \to \mathbb{R}$ is a nonparametric function, and Z_j represents noise.

- E.g., linear SEMs: $X_j = w_j^\top X + Z_j$, where $W = [w_1 \mid \cdots \mid w_d]$ represents the weighted adjacency matrix.

$$\begin{array}{cccc} X_1 & X_2 & X_3 & X_4 \\ \end{array}$$
$$\begin{bmatrix} 1.00 & -2.14 & 0.87 & -1.82 \\ -1.5 & 0.39 & 0.45 & -0.09 \\ \vdots & \vdots & \vdots & \vdots \end{bmatrix}$$



Problem introduction

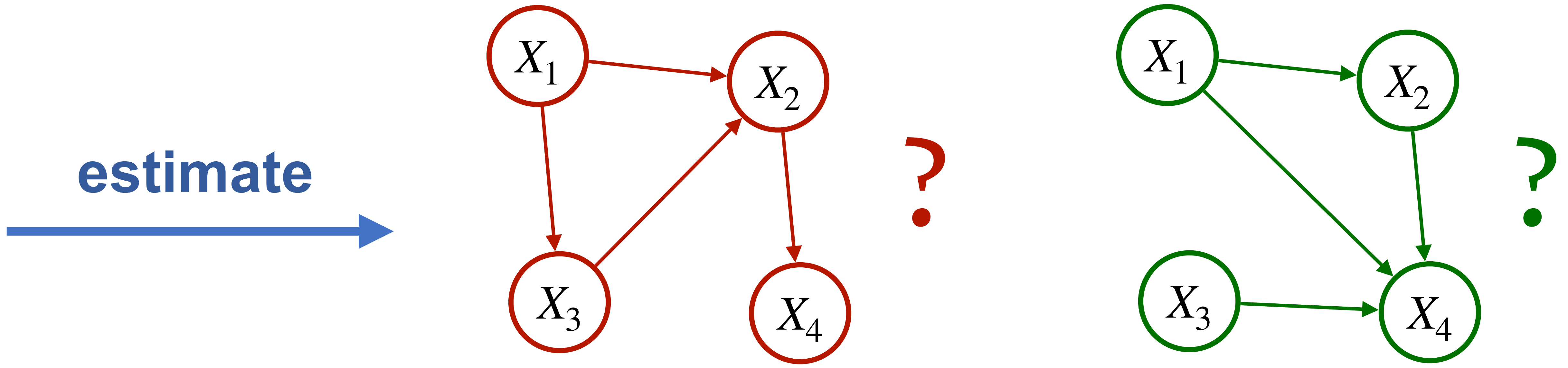
- The goal is to learn the underlying **directed acyclic graph (DAG)** of a structural equation model (SEM). A *Markovian* nonparametric SEM consists of a set of equations of the form,

$$X_j = f_j(X, Z_j),\ \forall j \in [d],$$

where each $f_j : \mathbb{R}^{d+1} \to \mathbb{R}$ is a nonparametric function, and Z_j represents noise.

- E.g., linear SEMs: $X_j = w_j^\top X + Z_j$, where $W = [w_1 \mid \cdots \mid w_d]$ represents the weighted adjacency matrix.

$$\begin{array}{cccc} X_1 & X_2 & X_3 & X_4 \end{array}$$
$$\begin{bmatrix} 1.00 & -2.14 & 0.87 & -1.82 \\ -1.5 & 0.39 & 0.45 & -0.09 \\ \vdots & \vdots & \vdots & \vdots \end{bmatrix}$$



Score-based approach

Score-based approach

A score-based method searches for the (weighted) adjacency matrix W that minimizes a given score Q that measures how well W fits the observed data $\mathbf{X}$. That is, we aim to solve

Score-based approach

A score-based method searches for the (weighted) adjacency matrix W that minimizes a given score Q that measures how well W fits the observed data $\mathbf{X}$. That is, we aim to solve

$$\min_{W} Q(W; \mathbf{X}) \quad \text{s.t.} \quad W \in \text{DAGs}\,.$$

Score-based approach

A score-based method searches for the (weighted) adjacency matrix W that minimizes a given score Q that measures how well W fits the observed data $\mathbf{X}$. That is, we aim to solve

$$\min_{W} Q(W; \mathbf{X}) \quad \text{s.t.} \quad W \in \text{DAGs} \, .$$

The above problem is known to be NP-complete to solve (Chickering 1996).

A continuous framework

A continuous framework

Recent work by Zheng et al. (2018) has replaced the combinatorial DAG constraint to a continuous constraint via the smooth function $h_{\text{expm}}(W) = \text{Tr}(e^{W \circ W}) - d$. That is,

A continuous framework

Recent work by Zheng et al. (2018) has replaced the combinatorial DAG constraint to a continuous constraint via the smooth function $h_{\text{expm}}(W) = \text{Tr}(e^{W \circ W}) - d$. That is,

$$\min_{W} Q(W; \mathbf{X}) \quad \text{s.t. } h_{\text{expm}}(W) = 0.$$

A continuous framework

Recent work by Zheng et al. (2018) has replaced the combinatorial DAG constraint to a continuous constraint via the smooth function $h_{\text{expm}}(W) = \text{Tr}(e^{W \circ W}) - d$. That is,

$$\min_{W} Q(W; \mathbf{X}) \quad \text{s.t. } h_{\text{expm}}(W) = 0.$$

The above is possible since $h_{\text{expm}}(W) = 0$ if and only if W is a DAG.

A new acyclicity characterization via log-determinant

A new acyclicity characterization via log-determinant

Motivated by the *nilpotency property* of DAGs (i.e., all eigenvalues of W are zero *if and only if* W is a DAG), we propose the following acyclicity characterization:

A new acyclicity characterization via log-determinant

Motivated by the *nilpotency property* of DAGs (i.e., all eigenvalues of W are zero *if and only if* W is a DAG), we propose the following acyclicity characterization:

$$h^s_{\mathrm{ldet}}(W) = -\log\det(sI - W \circ W) + d\log s\,.$$

A new acyclicity characterization via log-determinant

Motivated by the *nilpotency property* of DAGs (i.e., all eigenvalues of W are zero *if and only if* W is a DAG), we propose the following acyclicity characterization:

$$h^s_{\text{ldet}}(W) = -\log\det(sI - W \circ W) + d \log s \,.$$

To be a proper acyclicity function, we show that $sI - W \circ W$ must be an M-matrix, i.e., $\rho(W \circ W) < s$.

A new acyclicity characterization via log-determinant

Motivated by the *nilpotency property* of DAGs (i.e., all eigenvalues of W are zero *if and only if* W is a DAG), we propose the following acyclicity characterization:

$$h^s_{\text{ldet}}(W) = -\log\det(sI - W \circ W) + d \log s\,.$$

To be a proper acyclicity function, we show that $sI - W \circ W$ must be an M-matrix, i.e., $\rho(W \circ W) < s$.

Theorem 1 (Informal). *For any $s > 0$. The following holds:*

(i) $h^s_{\text{ldet}}(W) \geq 0$. Moreover, $h^s_{\text{ldet}}(W) = 0$ *if and only if* W is a DAG.

(ii) $\nabla h^s_{\text{ldet}}(W) = 2(sI - W \circ W)^{-\top} \circ W$. Moreover, $\nabla h^s_{\text{ldet}}(W) = 0$ *if and only if* W is a DAG.

Properties

Properties

- Its negative gradient points towards the interior of the set of *M*-matrices.

Properties

- Its negative gradient points towards the interior of the set of M-matrices.
- Has a **simpler and tractable** closed form expression of its **Hessian**.

Properties

- Its negative gradient points towards the interior of the set of M-matrices.
- Has a **simpler and tractable** closed form expression of its **Hessian**.
- Acts as a **regularizer**: shrinks the values of parameters that are part of a cycle. (all)

Properties

- Its negative gradient points towards the interior of the set of M-matrices.
- Has a **simpler and tractable** closed form expression of its **Hessian**.
- Acts as a **regularizer**: shrinks the values of parameters that are part of a cycle. (all)
- It is an **invex function**, i.e., all its stationary points are global minima (DAGs). (all)

Properties

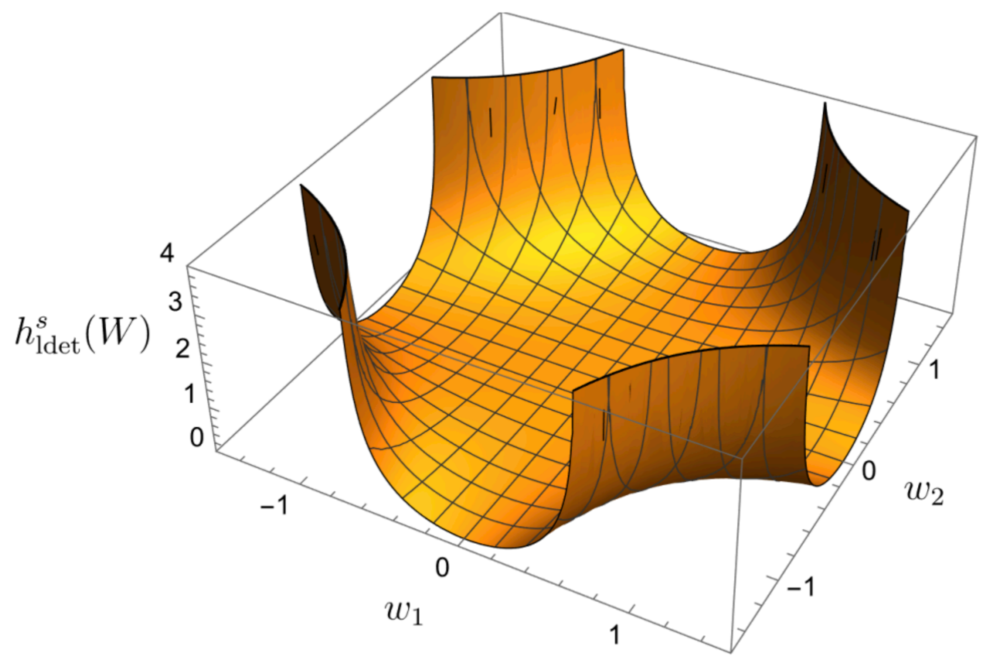
- Its negative gradient points towards the interior of the set of M-matrices.
- Has a **simpler and tractable** closed form expression of its **Hessian**.
- Acts as a **regularizer**: shrinks the values of parameters that are part of a cycle. (all)
- It is an **invex function**, i.e., all its stationary points are global minima (DAGs). (all)

$$W = \begin{bmatrix} 0 & w_1 \\ w_2 & 0 \end{bmatrix}$$

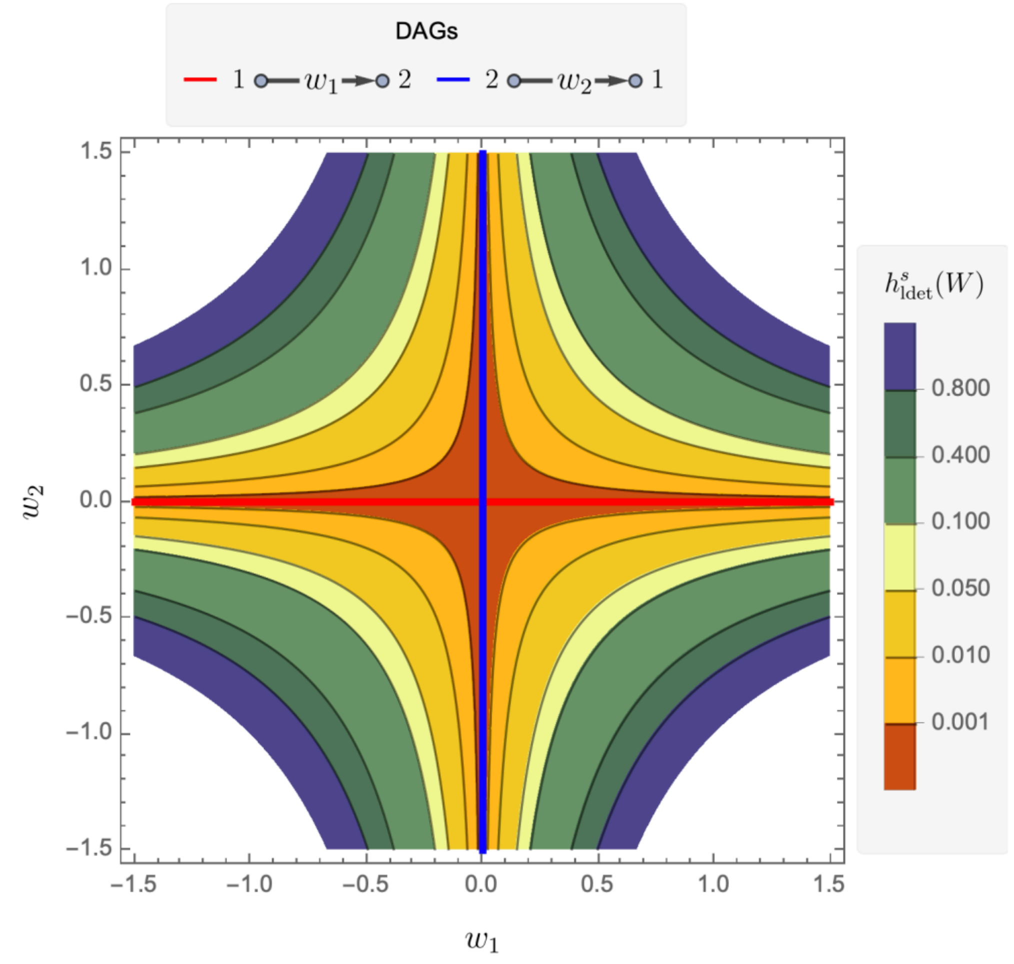
Properties

- Its negative gradient points towards the interior of the set of M-matrices.
- Has a **simpler and tractable** closed form expression of its **Hessian**.
- Acts as a **regularizer**: shrinks the values of parameters that are part of a cycle. (all)
- It is an **invex function**, i.e., all its stationary points are global minima (DAGs). (all)

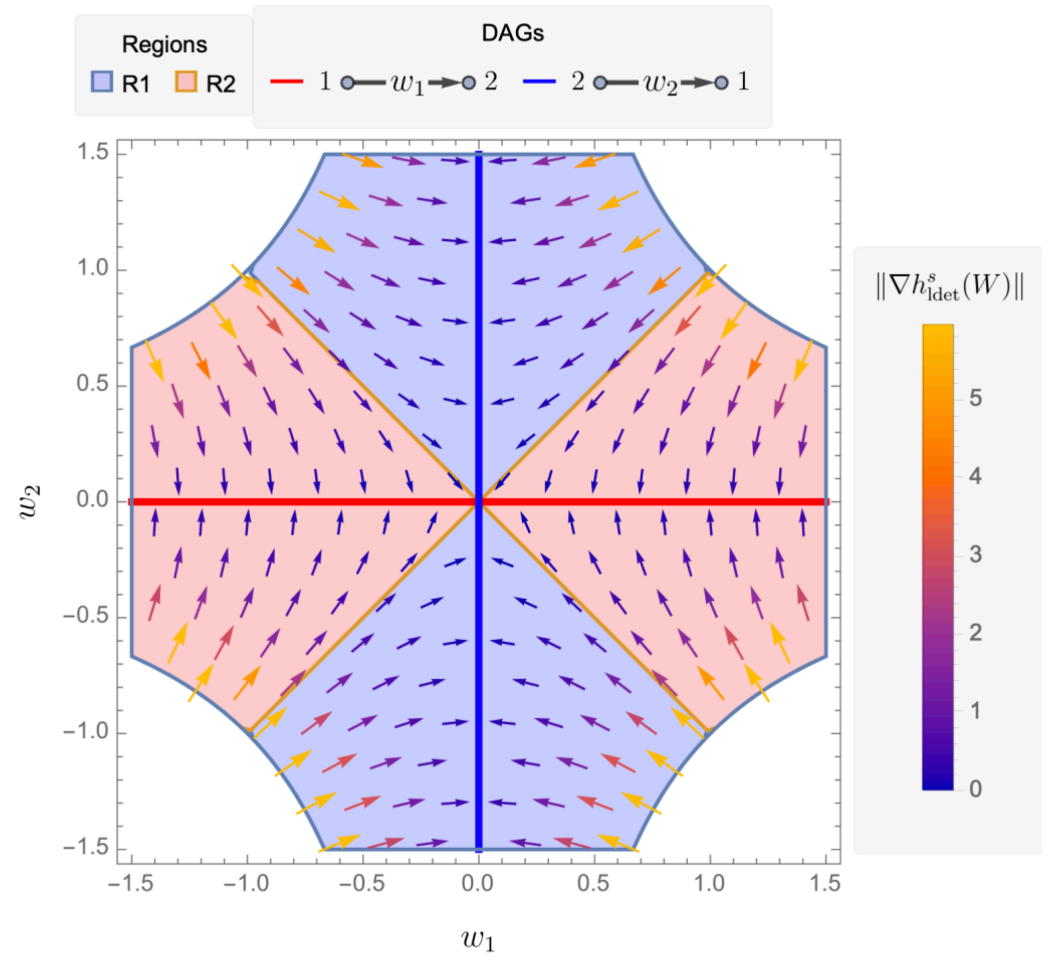
$$W = \begin{bmatrix} 0 & w_1 \\ w_2 & 0 \end{bmatrix}$$



(a) $h^{s=1}_{\mathrm{ldet}}(W)$



(b) Contours of $h^{s=1}_{\mathrm{ldet}}(W)$



(c) Vector field of $\nabla h^{s=1}_{\mathrm{ldet}}(W)$

Benefits of using the log-determinant

Benefits of using the log-determinant

- Does not diminish cycles of any length. In contrast, h_{expm} diminishes a cycle of length k by $1/k!$.

Benefits of using the log-determinant

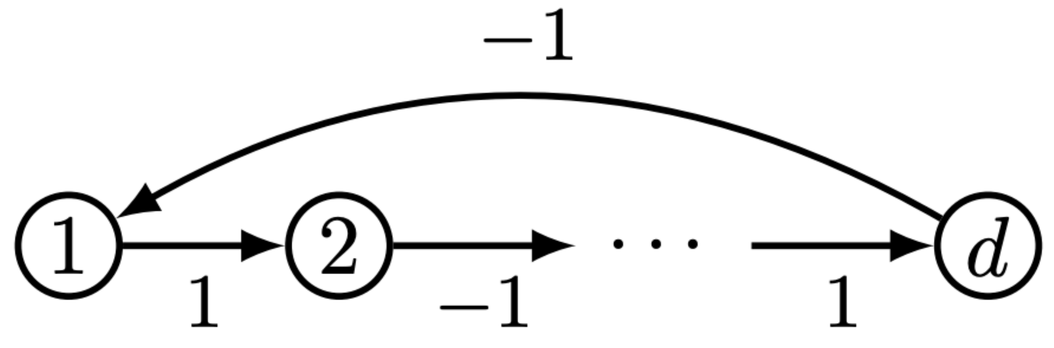
- Does not diminish cycles of any length. In contrast, h_{expm} diminishes a cycle of length k by $1/k!$.
- Has better behaved gradients.

Benefits of using the log-determinant

- Does not diminish cycles of any length. In contrast, h_{expm} diminishes a cycle of length k by $1/k!$.
- Has better behaved gradients.
- Computing h^s_{ldet} is empirically faster than h_{expm} and h_{poly}.

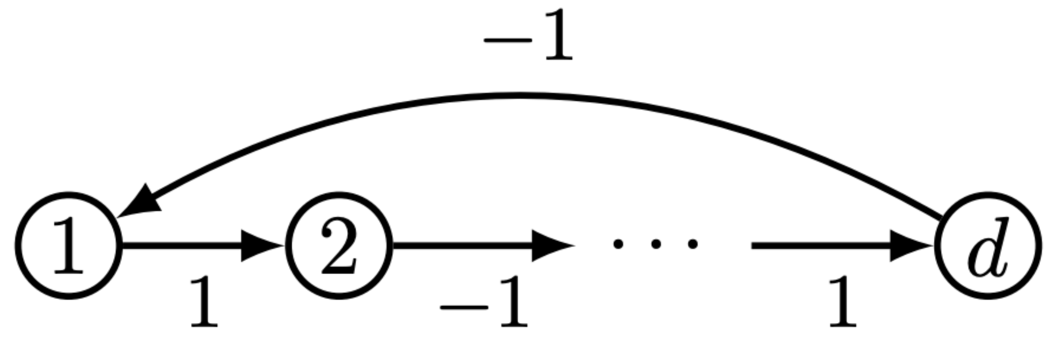
Benefits of using the log-determinant

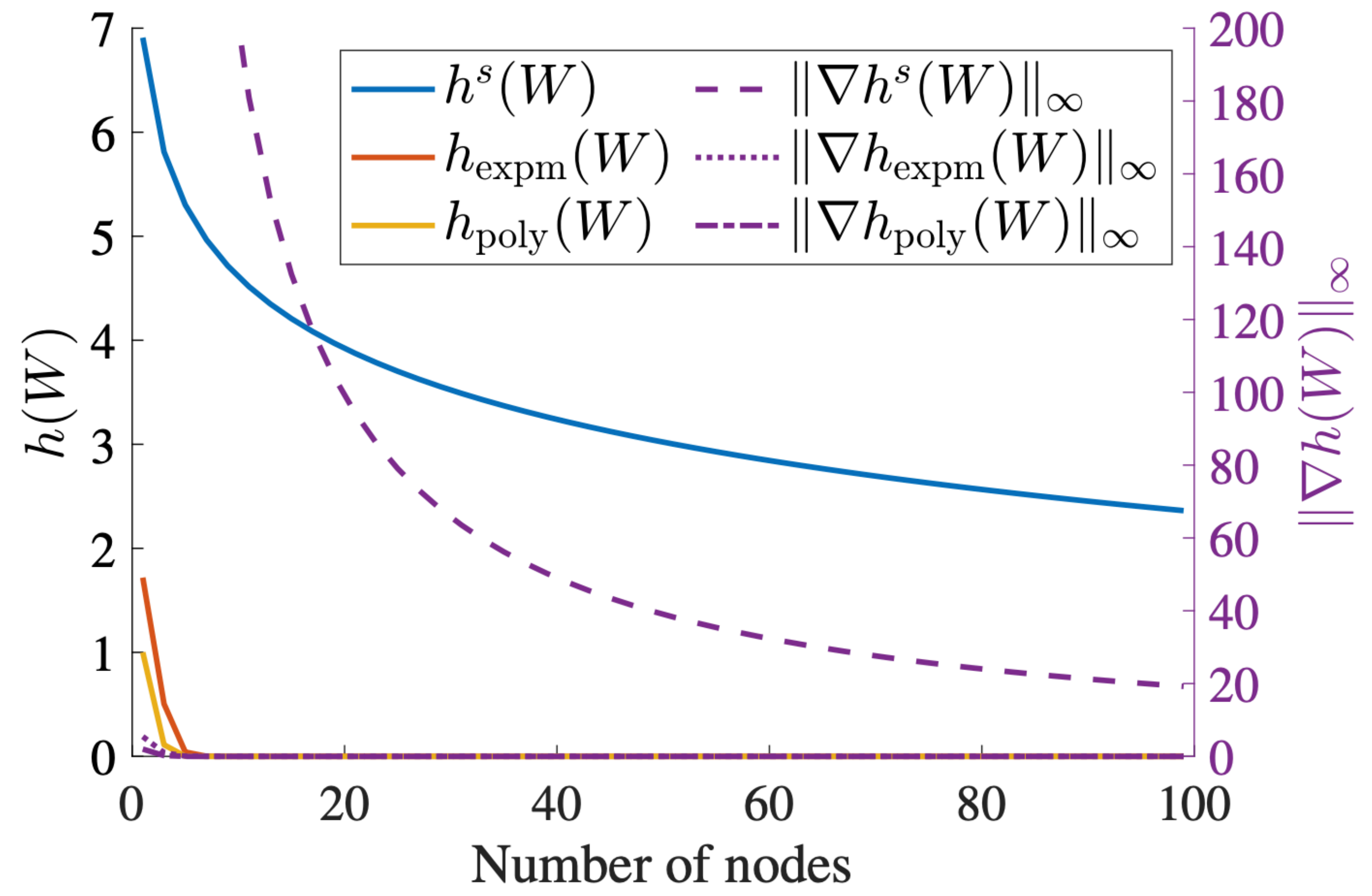
- Does not diminish cycles of any length. In contrast, h_{expm} diminishes a cycle of length k by $1/k!$.
- Has better behaved gradients.
- Computing h^s_{ldet} is empirically faster than h_{expm} and h_{poly}.



Benefits of using the log-determinant

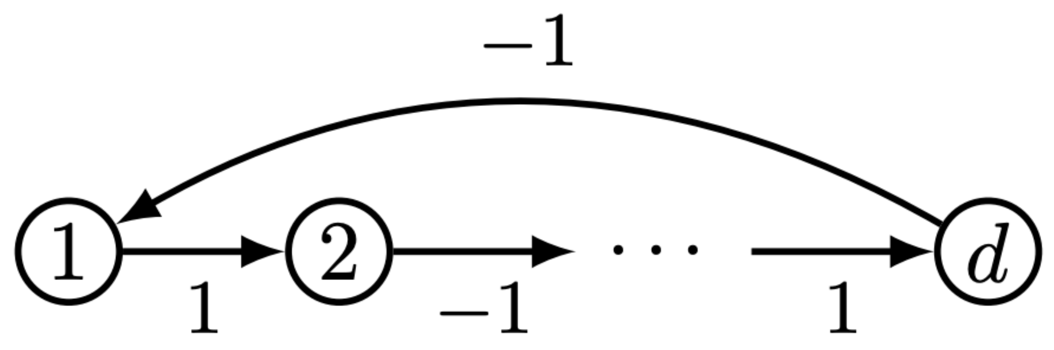
- Does not diminish cycles of any length. In contrast, h_{expm} diminishes a cycle of length k by $1/k!$.
- Has better behaved gradients.
- Computing h^s_{ldet} is empirically faster than h_{expm} and h_{poly}.

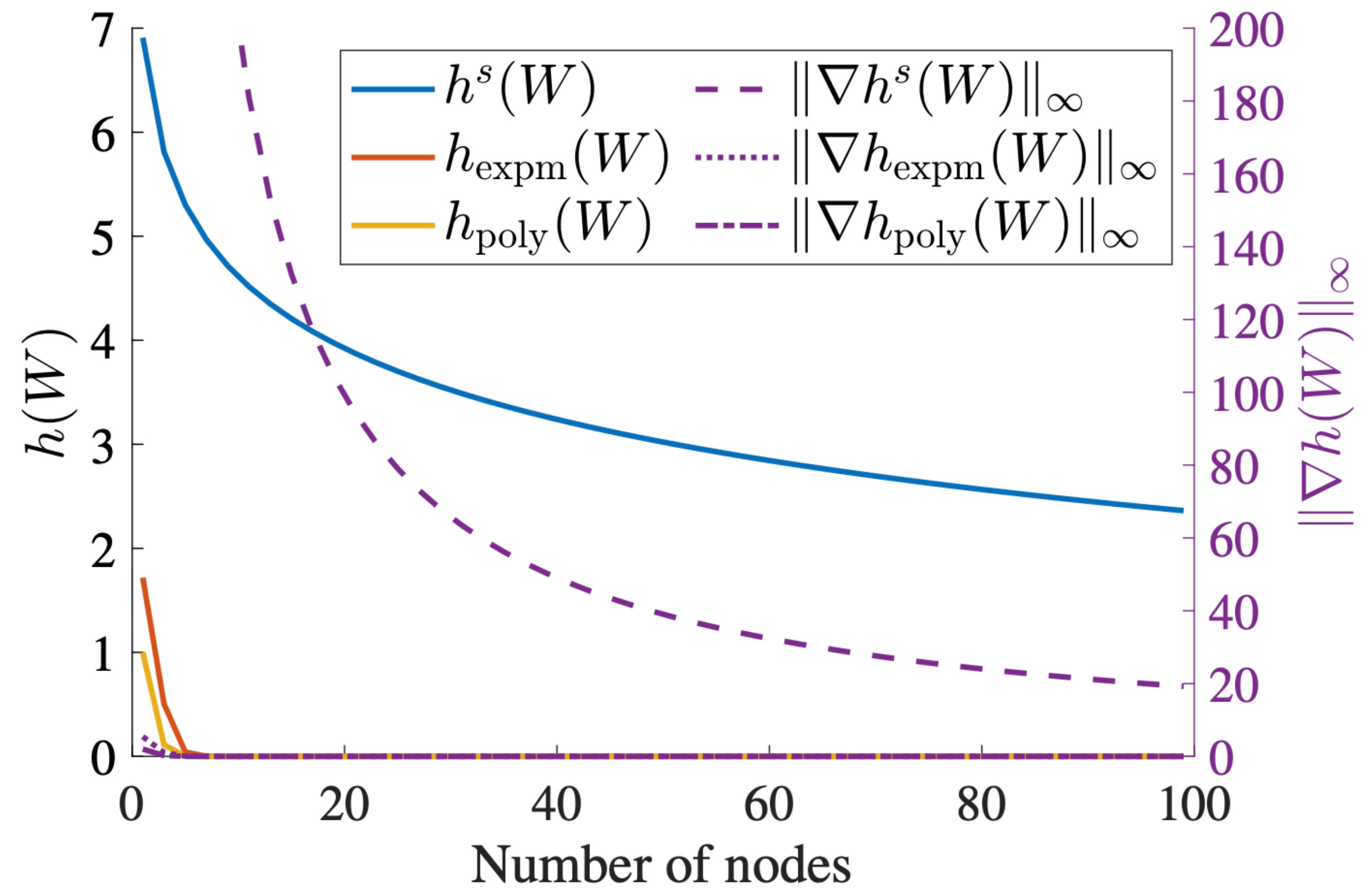


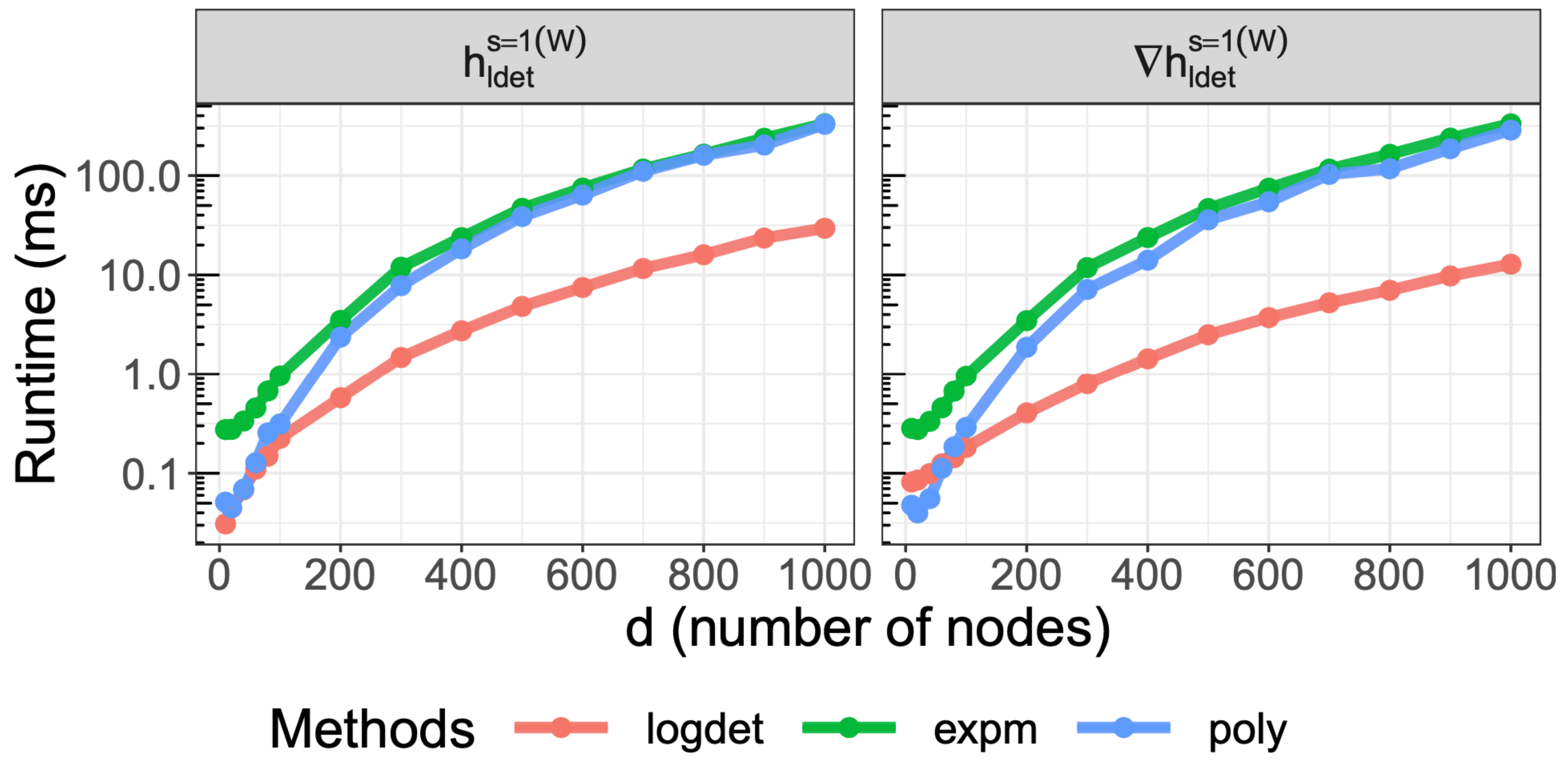


Benefits of using the log-determinant

- Does not diminish cycles of any length. In contrast, h_{expm} diminishes a cycle of length k by $1/k!$.
- Has better behaved gradients.
- Computing h^s_{ldet} is empirically faster than h_{expm} and h_{poly}.







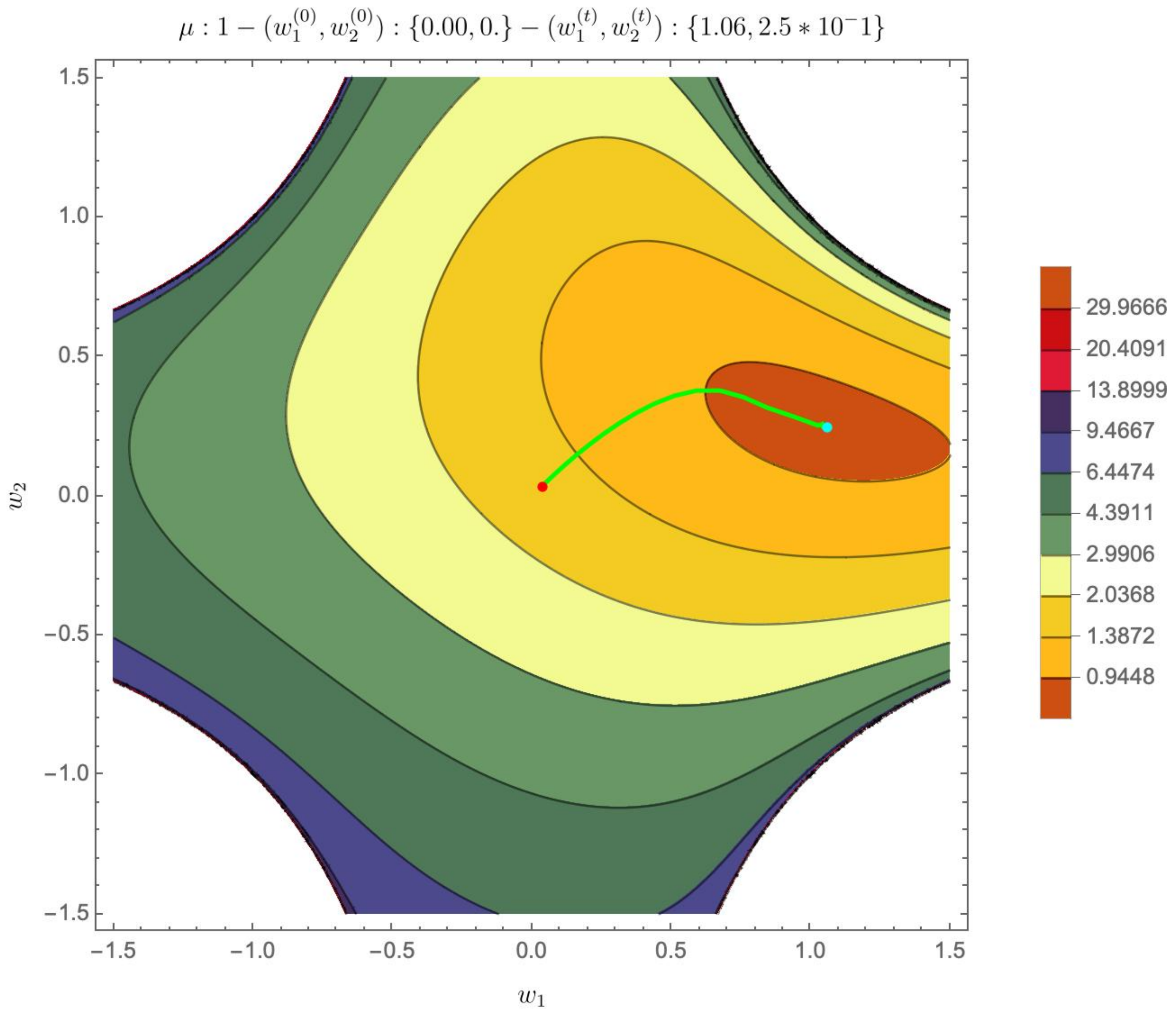
Optimization

Optimization

$$W_{\text{true}} = \begin{bmatrix} 0 & 1.2 \\ 0 & 0 \end{bmatrix}$$

Optimization

$$W_{\text{true}} = \begin{bmatrix} 0 & 1.2 \\ 0 & 0 \end{bmatrix}$$



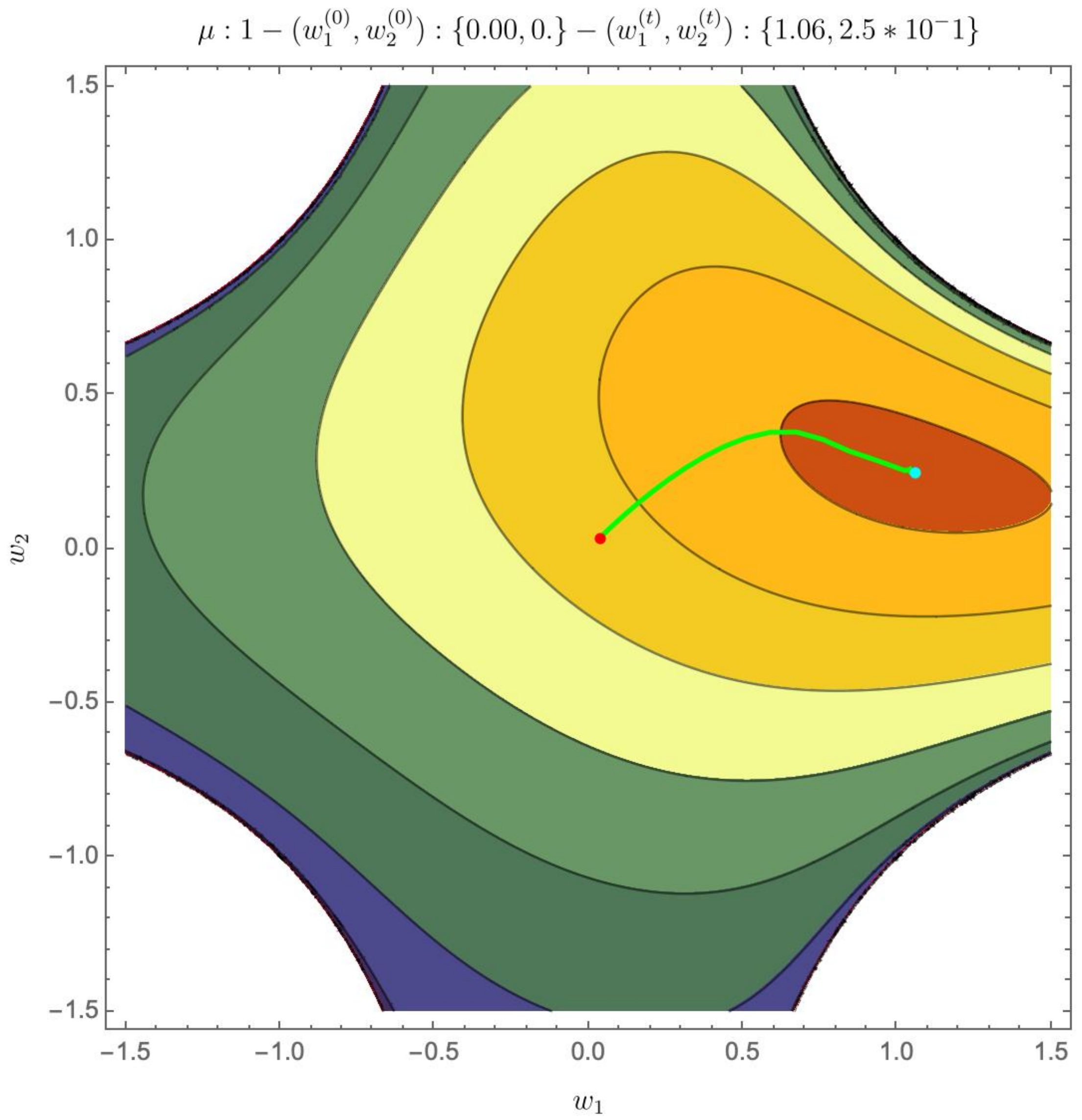
$$\min_{W} \boxed{1} \cdot Q(W) + h(W)$$

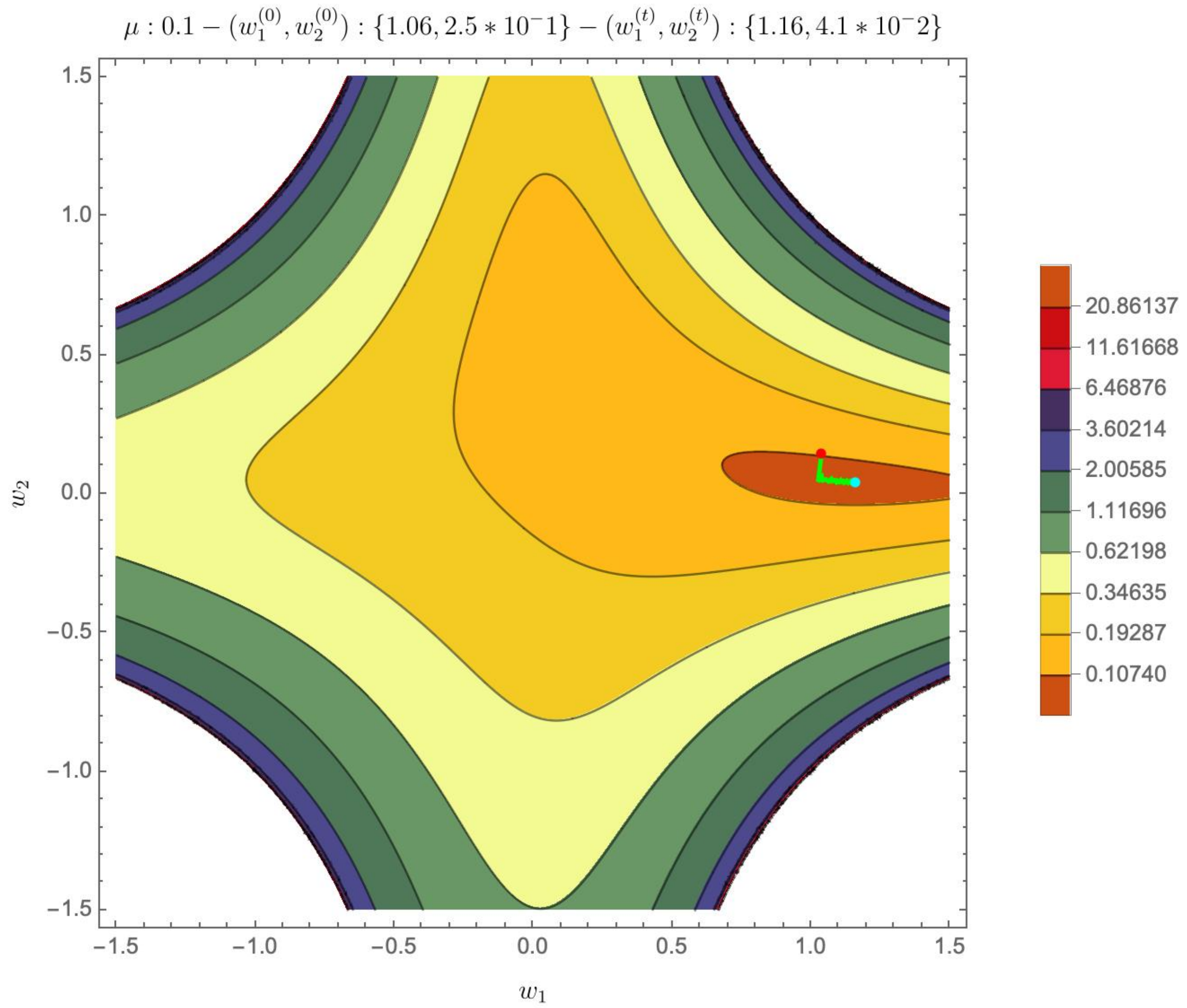
$$W_{\text{init}} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$$

$$W_{\text{sol}} = \begin{bmatrix} 0 & 1.06 \\ 0.25 & 0 \end{bmatrix}$$

Optimization

$$W_{\text{true}} = \begin{bmatrix} 0 & 1.2 \\ 0 & 0 \end{bmatrix}$$





$$\min_W 1 \cdot Q(W) + h(W)$$

$$W_{\text{init}} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$$

$$W_{\text{sol}} = \begin{bmatrix} 0 & 1.06 \\ 0.25 & 0 \end{bmatrix}$$

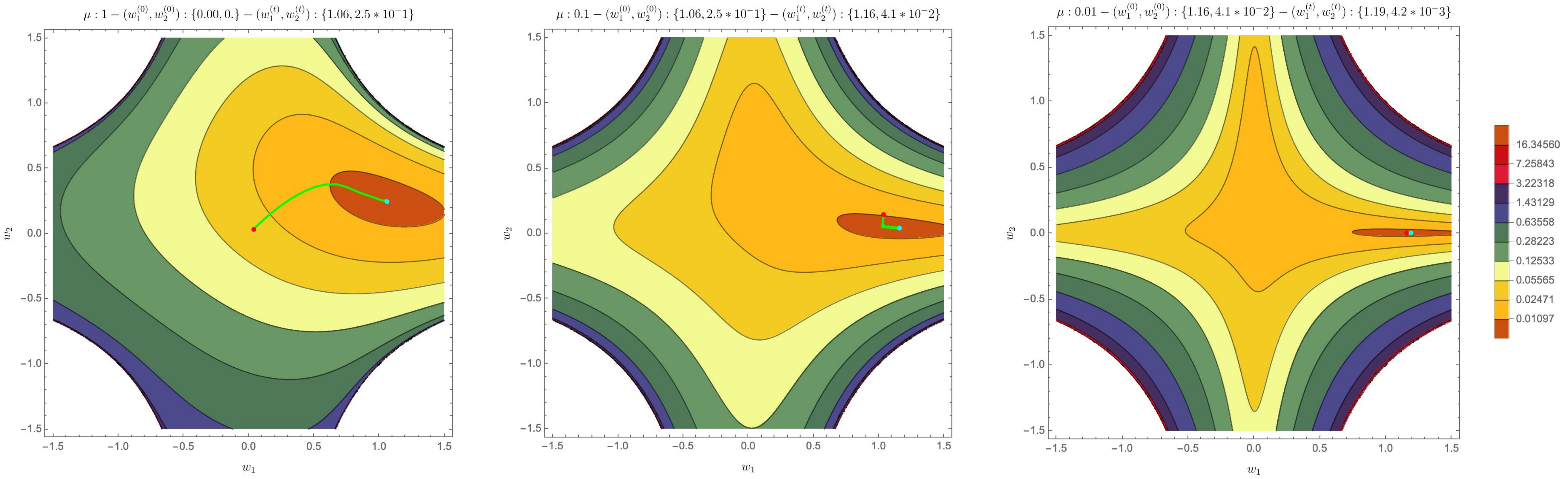
$$\min_W 0.1 \cdot Q(W) + h(W)$$

$$W_{\text{init}} = \begin{bmatrix} 0 & 1.06 \\ 0.25 & 0 \end{bmatrix}$$

$$W_{\text{sol}} = \begin{bmatrix} 0 & 1.16 \\ 0.041 & 0 \end{bmatrix}$$

Optimization

$$W_{\text{true}} = \begin{bmatrix} 0 & 1.2 \\ 0 & 0 \end{bmatrix}$$



$$\min_W \boxed{1} \cdot Q(W) + h(W)$$

$$W_{\text{init}} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$$

$$W_{\text{sol}} = \begin{bmatrix} 0 & 1.06 \\ 0.25 & 0 \end{bmatrix}$$

$$\min_W \boxed{0.1} \cdot Q(W) + h(W)$$

$$W_{\text{init}} = \begin{bmatrix} 0 & 1.06 \\ 0.25 & 0 \end{bmatrix}$$

$$W_{\text{sol}} = \begin{bmatrix} 0 & 1.16 \\ 0.041 & 0 \end{bmatrix}$$

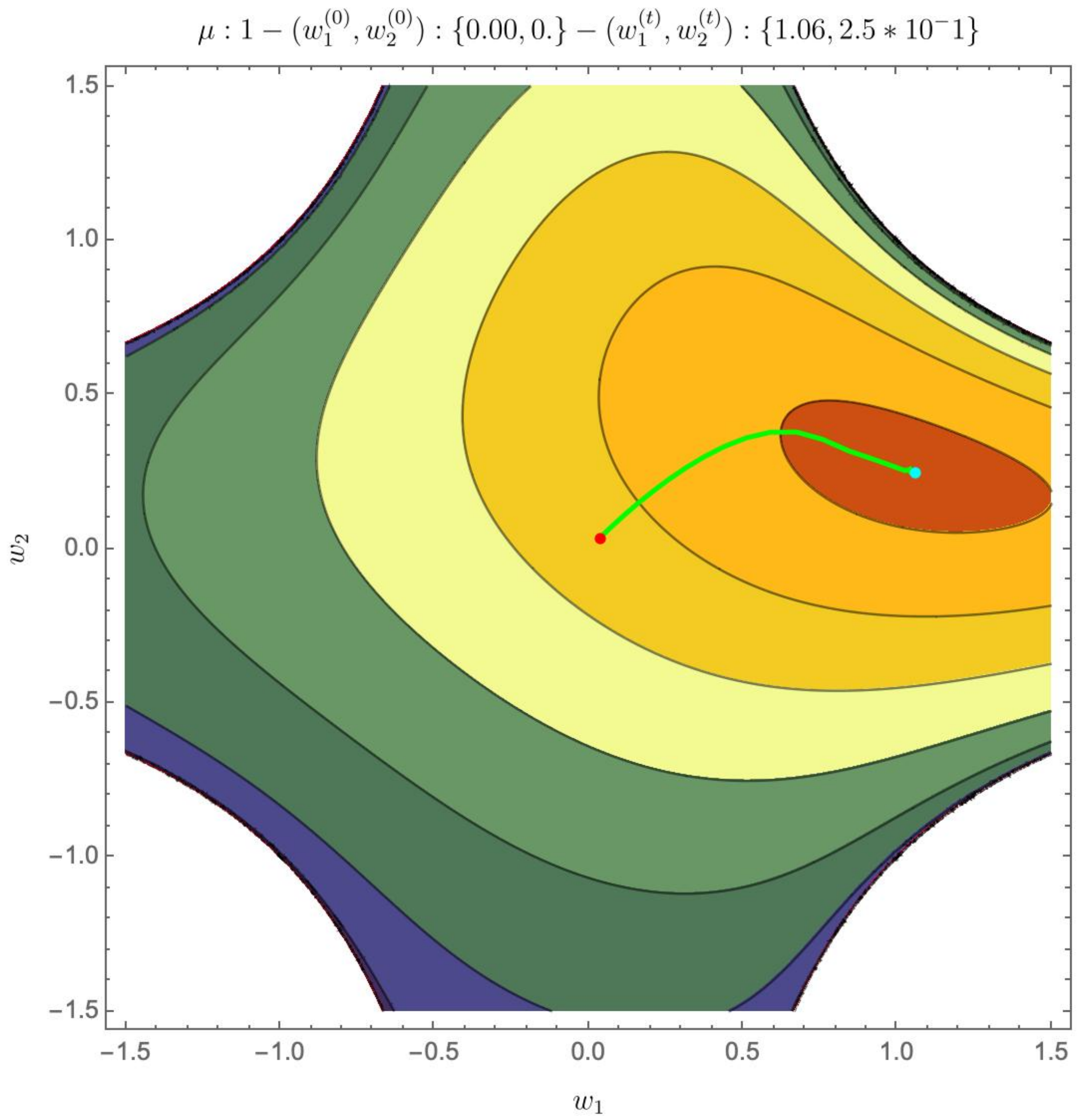
$$\min_W \boxed{0.01} \cdot Q(W) + h(W)$$

$$W_{\text{init}} = \begin{bmatrix} 0 & 1.16 \\ 0.041 & 0 \end{bmatrix}$$

$$W_{\text{sol}} = \begin{bmatrix} 0 & 1.19 \\ 0.0042 & 0 \end{bmatrix}$$

Optimization

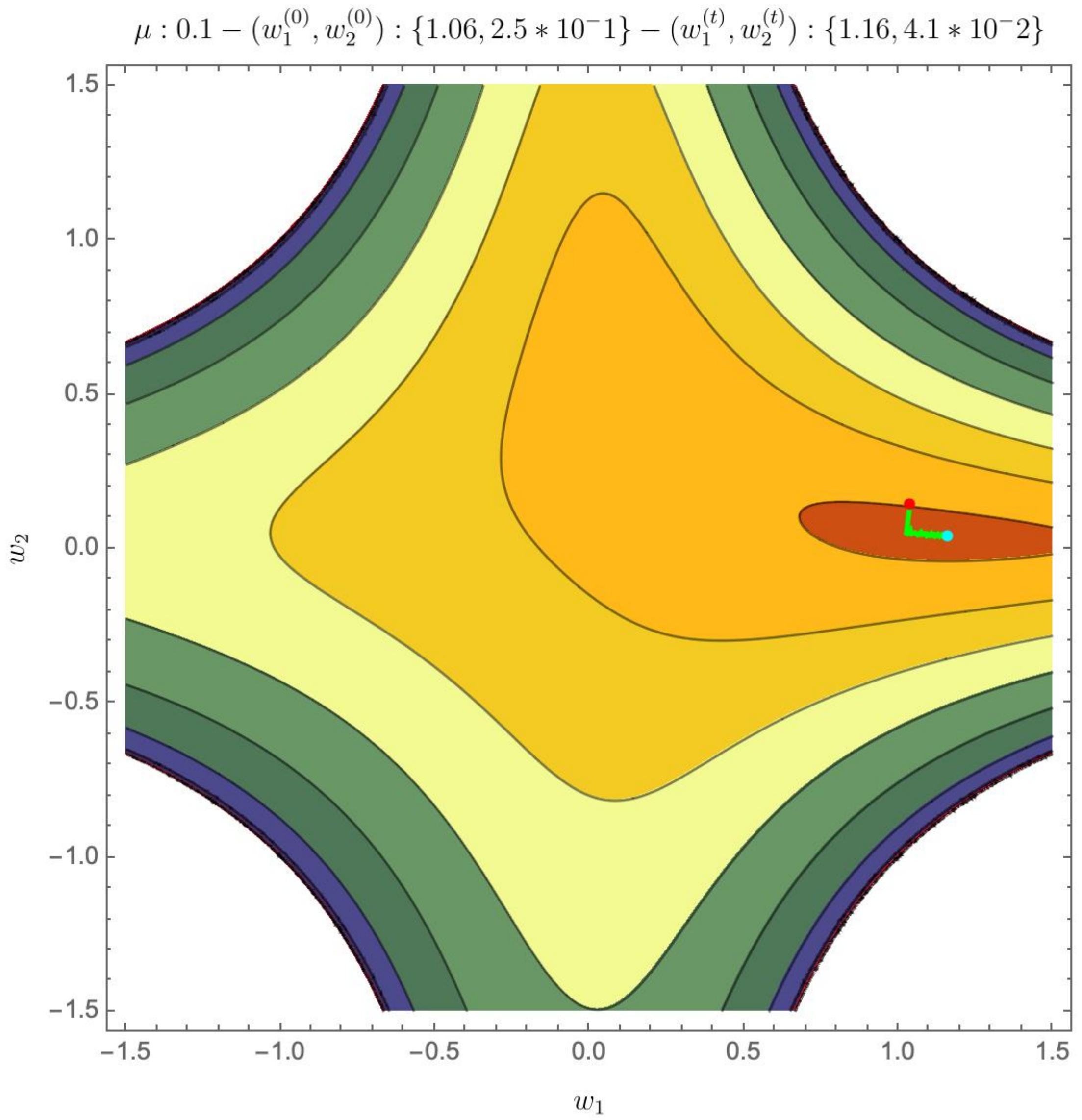
$$W_{\text{true}} = \begin{bmatrix} 0 & 1.2 \\ 0 & 0 \end{bmatrix}$$



$$\min_W \boxed{1} \cdot Q(W) + h(W)$$

$$W_{\text{init}} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$$

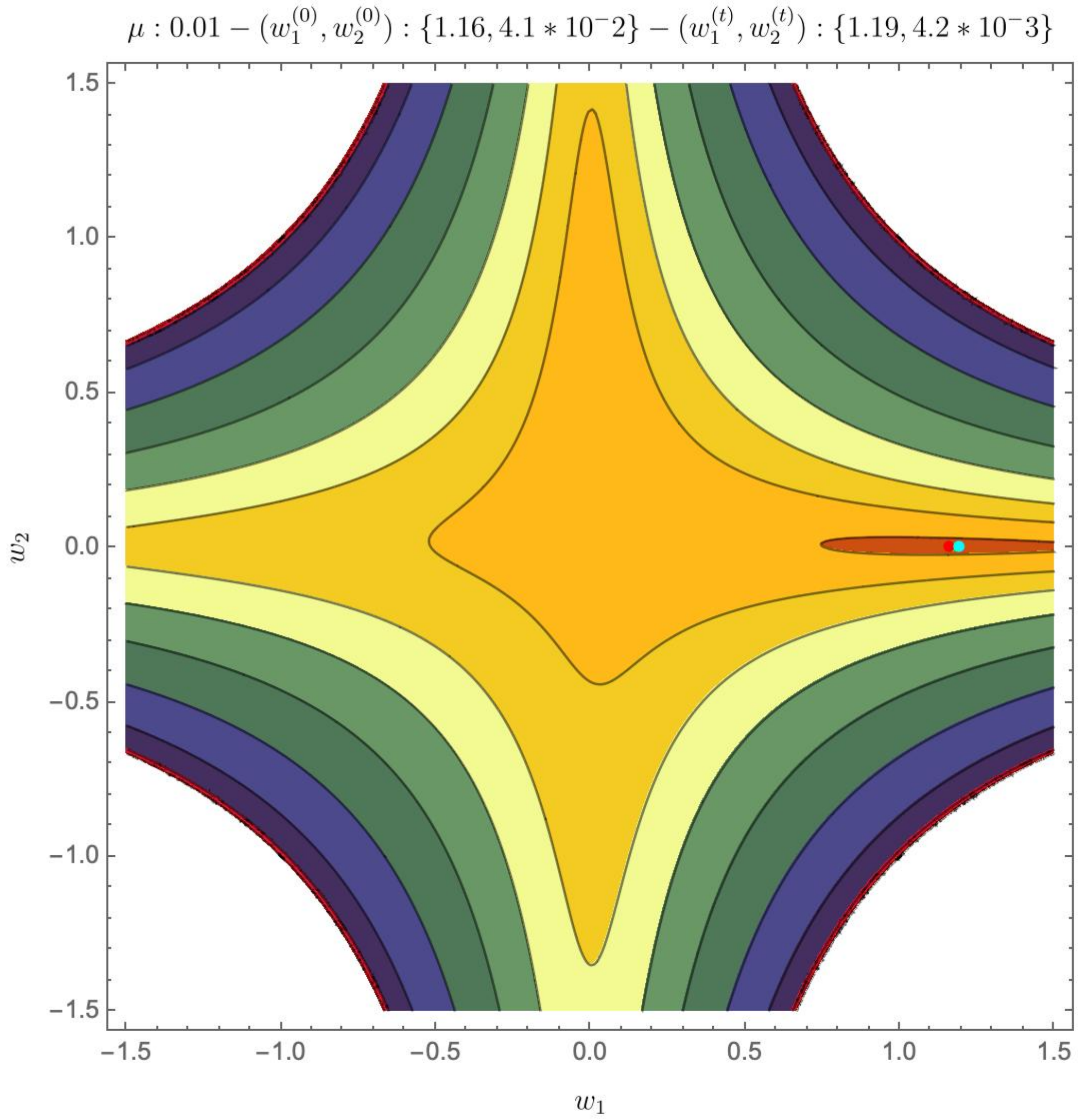
$$W_{\text{sol}} = \begin{bmatrix} 0 & 1.06 \\ 0.25 & 0 \end{bmatrix}$$



$$\min_W \boxed{0.1} \cdot Q(W) + h(W)$$

$$W_{\text{init}} = \begin{bmatrix} 0 & 1.06 \\ 0.25 & 0 \end{bmatrix}$$

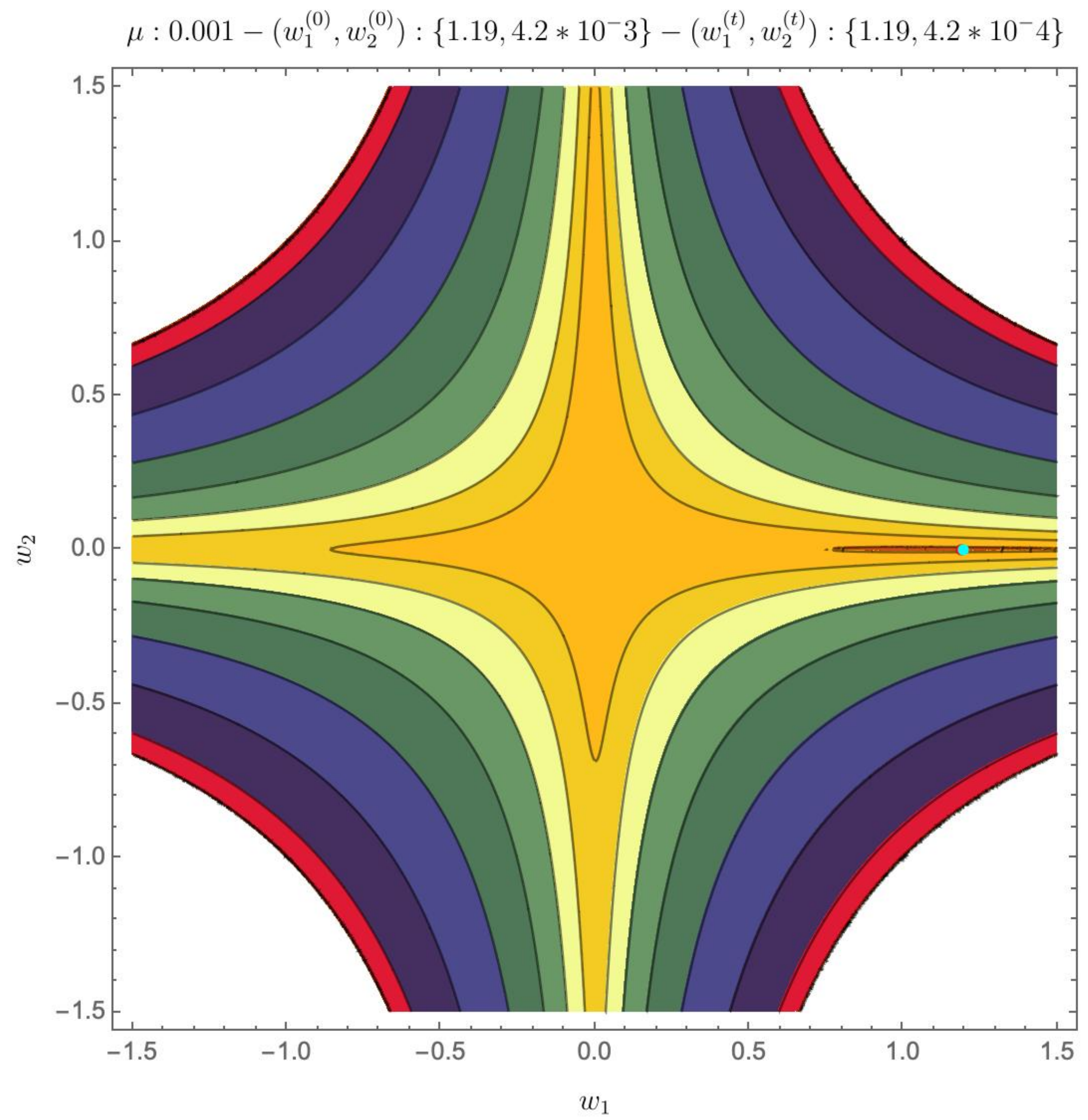
$$W_{\text{sol}} = \begin{bmatrix} 0 & 1.16 \\ 0.041 & 0 \end{bmatrix}$$



$$\min_W \boxed{0.01} \cdot Q(W) + h(W)$$

$$W_{\text{init}} = \begin{bmatrix} 0 & 1.16 \\ 0.041 & 0 \end{bmatrix}$$

$$W_{\text{sol}} = \begin{bmatrix} 0 & 1.19 \\ 0.0042 & 0 \end{bmatrix}$$



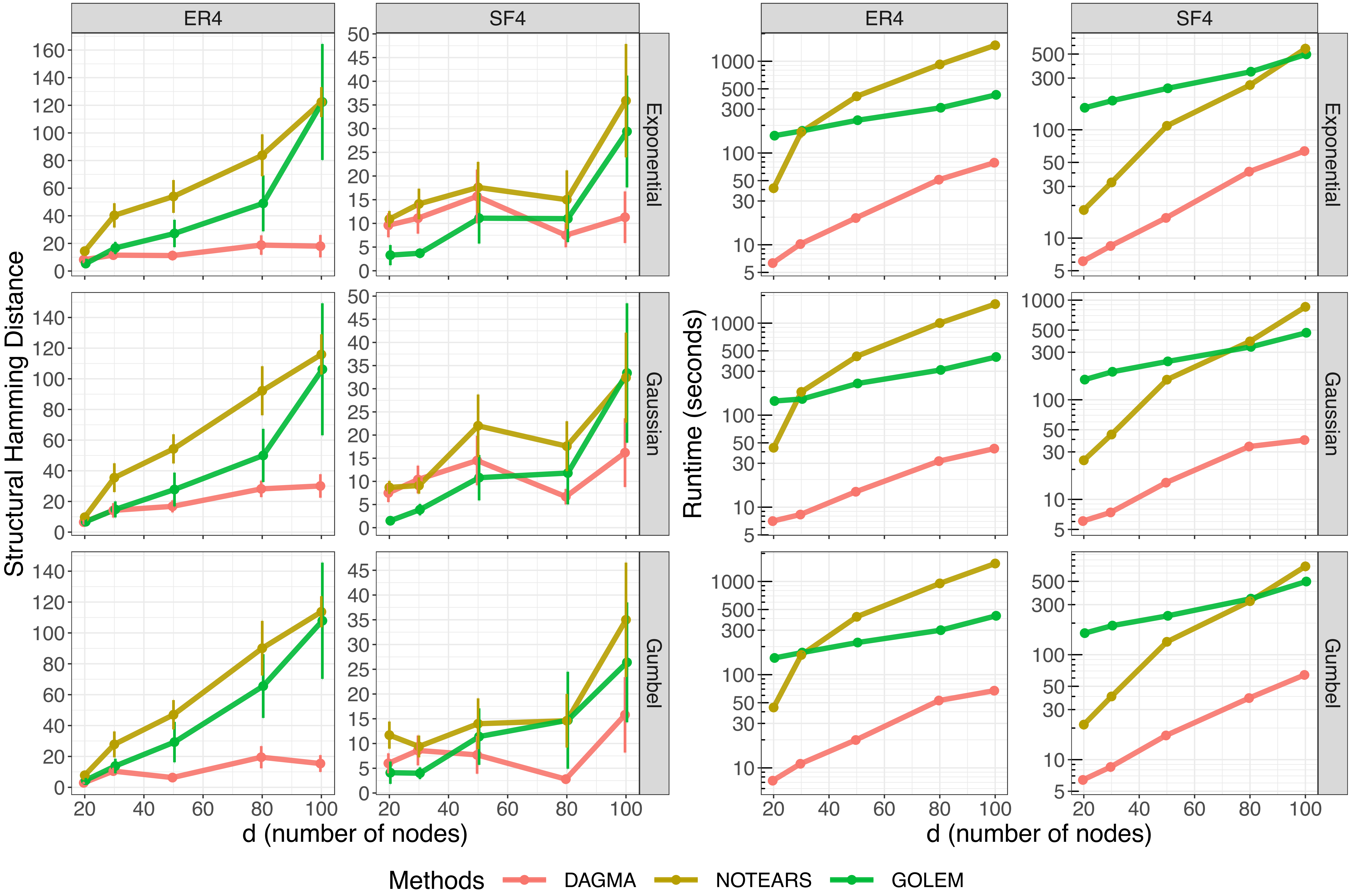
$$\min_W \boxed{0.001} \cdot Q(W) + h(W)$$

$$W_{\text{init}} = \begin{bmatrix} 0 & 1.19 \\ 0.0042 & 0 \end{bmatrix}$$

$$W_{\text{init}} = \begin{bmatrix} 0 & 1.19 \\ 0.00042 & 0 \end{bmatrix}$$

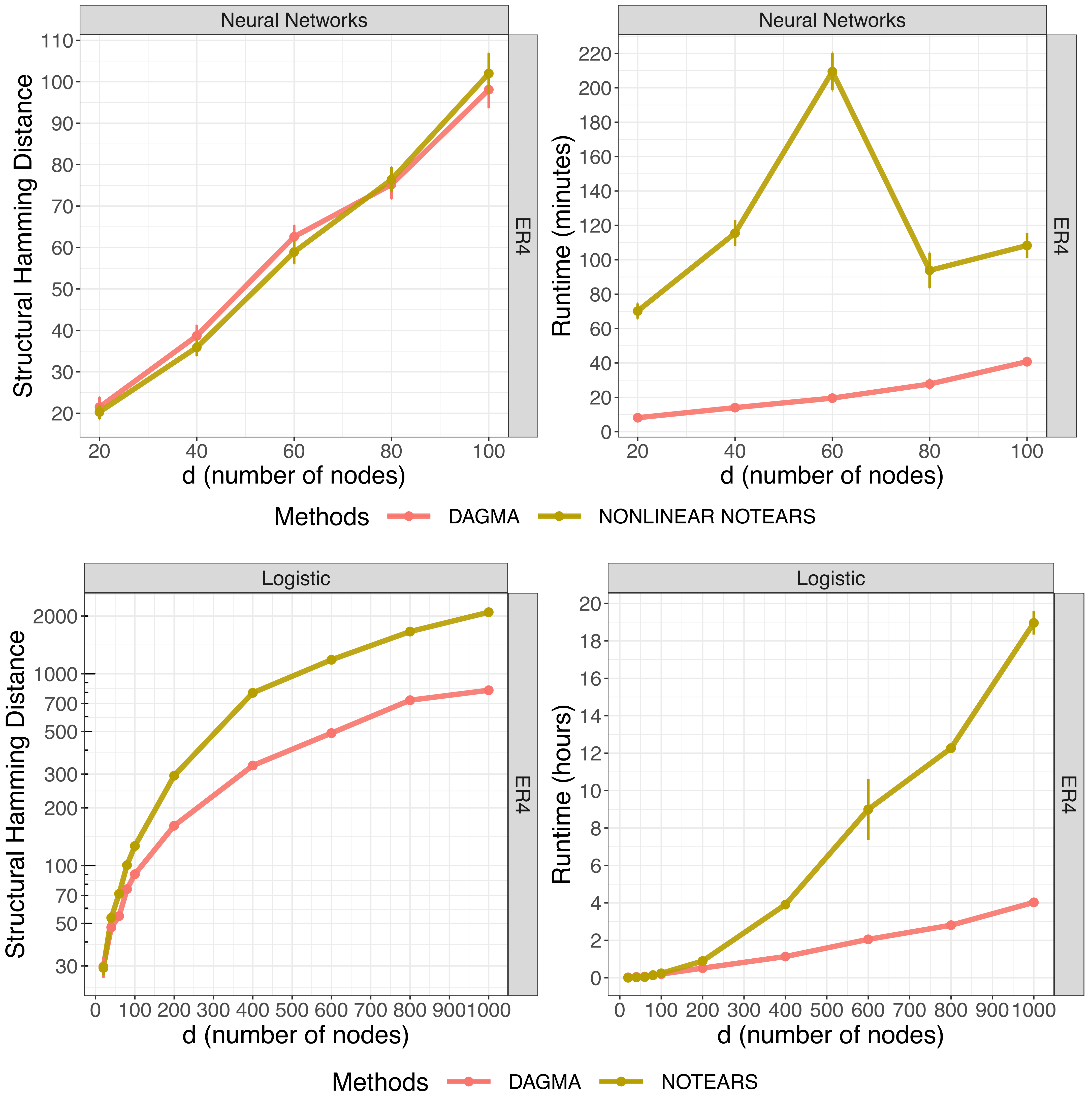
Empirical improvements

Linear SEMs



Empirical improvements

Nonlinear SEMs



Future directions

Future directions

- Exploit the Hessian structure of the log-det function for faster second-order methods.

Future directions

- Exploit the Hessian structure of the log-det function for faster second-order methods.
- In general, there is a need for rigorous guarantees of these continuous approaches:
  - Identifiability
  - Statistical/Computational guarantees