

理論談話会 vol1

End to end learning of user equilibrium with Implicit Neural Network (INN)

Zhichen Liu , Yafeng Yin *, Fan Bai , Donald K. Grimm
Transportation Research Part C 150 (2023) 104085

【新規性】

1. 「ボトムアップ」アプローチからの脱出

- ・ 行動モデルそのものをデータから学習「end to end」の枠組みを提案
- ・ 行動モデルの選択と均衡の導出を一体化

2. 暗黙層（implicit layer）としてのVIの導入

- ・ 交通均衡条件をNNに基づく変分不等式（VI）として表現
- ・ NNとゲーム理論的均衡を統合するアーキテクチャは前例が少ない

3. ネットワーク構造変更に強いニューラルネット設計

- ・ 次元を揃えて出力することで、リンク数やノード数が変更されても再学習なしで利用可能なモデル構造を提案

【信頼性】

1. 均衡解の存在性を理論的に保証

- ・ 提案NN構造が連続性・単調性・Lipschitz性を満たす設計で均衡解の存在が定理で保証

2. 解法の安定性と収束性

- ・ オペレーター分割法 + Broyden法 + 正則化でバッチ単位で安定してVIを解く
- ・ 収束性も理論的に議論（Appendixに詳細あり）

3. 数値実験による検証

- ・ Sioux Fallsネットワークを用いたシミュレーションで既往手法(DUE, SUE)と比較し一貫して高精度

目次

1 : Introduction

2 : Back Ground

3 : New Methodology

3.1 : Neural-network-based VI formulation of UE (ニューラルネットワークに基づくUEの変分不等式定式化)

3.2 : End-to-end learning with an implicit layer (インプリシット層を用いたエンドツーエンド学習)

4 : Experiments

5 : Result

6 : Conclusion

1 : Introduction

研究の背景

ユーザー均衡の重要性と伝統的手法の限界

交通ネットワーク均衡モデリングのこれまで

Step 1: 行動モデルの設定（経路選択のルール）

- 最短経路選択 (Deterministic)
- 確率的選択 (Stochastic, Logit model)
- Generalized cost (時間 + 金銭 + 快適性 など)

Step 3: 経路集合の生成（経路集合行列）

- 各ODペアに対し候補経路を生成
- ODペア毎の経路集合マトリックス(P_{od})

Step 5: 数理モデル化（均衡条件の定式化）

- Wardropの第1原則に基づいた数理モデル表現
- 数理計画問題(MPECや凸最適化)

Step 2: 交通需要の設定（OD行列）

- 各ODペアに対し交通需要量を計算
- ODペアの需要量マトリックス(q_{od})

Step 4: 費用関数（リンク or 経路の時間関数）の定義

- 各リンク/経路 における通行コストを定義
BPR関数などを経路に設定

Step 6: 数値解法による均衡流の導出

- Frank-Wolfe法
- 固定点反復法(リンクベースの形式)
- プロジェクション法/オペレータ分割法(変分不等式)
- Path Equilibration法 など

研究の背景

ユーザー均衡の重要性と伝統的手法の限界

交通ネットワーク均衡モデリングのこれまで

Step 1: 行動モデルの設定（経路選択のルール）

- 最短経路選択 (Deterministic)
- 確率的選択 (Stochastic, Logit model)
- Generalized cost (時間 + 金銭 + 快適性 など)

Step 3: 経路集合の生成（経路集合行列）

- 各ODペアに対し候補経路を生成
- ODペア毎の経路集合マトリックス(P_{od})

Step 5: 数理モデル化（均衡条件の定式化）

- Wardropの第1原則に基づいた数理モデル表現
- 数理計画問題(MPECや凸最適化)

Step 2: 交通需要の設定（OD行列）

- 各ODペアに対し交通需要量を計算
- ODペアの需要量マトリックス(q_{od})

Step 4: 費用関数（リンク or 経路の時間関数）の定義

- 各リンク/経路 における通行コストを定義
BPR関数などを経路に設定

Step 6: 数値解法による均衡流の導出

- Frank-Wolfe法
- 固定点反復法(リンクベースの形式)
- プロジェクション法/オペレータ分割法
- Path Equilibration法 など

2 : Back Ground

研究の背景

ユーザー均衡の重要性と伝統的手法の限界

交通ネットワーク均衡モデリングのこれまで

年代	著者	タイトル	アプローチ	備考
1952	Wardrop	<i>Some theoretical aspects of road traffic research</i>	(原理)	Wardrop均衡の原理（第一法則）を初めて提案
1956	Beckmann, McGuire & Winsten	<i>Studies in the Economics of Transportation</i>	数理最適化	Wardrop均衡を凸最適化問題として定式化（Beckmannモデル）
1956	Frank & Wolfe	<i>An algorithm for quadratic programming</i>	数理最適化	Beckmannモデルを解くための効率的な手法（Frank-Wolfe法）
1979	Smith, M. J.	<i>The existence, uniqueness and stability of traffic equilibria</i>	変分不等式 (VI)	VI形式でユーザー均衡を定式化、非凸な費用関数や一般的ネットワークにも適用可能
1985	Sheffi, Y.	<i>Urban Transportation Networks: Equilibrium Analysis with Mathematical Programming Methods</i>	数理最適化	実務・教育でも使われる標準的な教科書、KKT条件・最適性条件にも着目（MPECの基礎）
1993	Nagurney, A.	<i>Network Economics: A Variational Inequality Approach</i>	変分不等式 (VI)	ネットワーク理論全般へのVIの応用を体系化(経済学における均衡全般)
1994	Patriksson, M.	<i>The Traffic Assignment Problem: Models and Methods</i>	理論体系の整理	交通割当モデルのモデリング～数値解法までを網羅 VI, KKT, LCP（線形相補問題）など形式間の対応整理
2002-2010	Facchinei & Pang / Ferris et al.	<i>Finite-Dimensional VI and MPEC Theory</i>	MPEC / KKT / LCP / MCP	MPEC（数理計画＋均衡制約）を体系化 VIを制約条件として組み込んだ最適化（MPEC）の理論的展開
2020	Li et al.	<i>Traffic flow estimation with neural networks: A variational inequality approach</i>	VI + ニューラルネット	経路コストをNNで推定しVIに埋め込む先駆的試み（行動モデルは仮定）
2023	Liu et al.	<i>End-to-end learning of user equilibrium with implicit neural networks</i>	VI + NN (Implicit layer)	行動モデルを仮定せずにNNで直接学習。解が VI の固定点として定義される implicit layerを用い、暗黙的に均衡制約を満たす

研究の背景

ユーザー均衡の重要性と伝統的手法の限界

交通ネットワーク均衡モデリングのこれまで

数理計画問題(MPECや凸最適化) / 変分不等式(VI)

	凸最適化 (数理計画)	変分不等式 (VI)
モデル 例	Beckmannモデル(Wardrop均衡を凸最適化問題で定式化)	Wardrop均衡の一般形 (非対称・非加法的コスト含む)
成立条件	対称性・可積分性・リンクアドディティブ	非対称性・非加法性 (例: 交差点の待ち時間や快適性) も含められる
解法	最適化手法 (Frank-Wolfeなど)	プロジェクション法、オペレーター分割
モデル条件	対称かつ加法的なリンクコストに依存	モノトン性があれば一般化可能 (リンク非加法・非対称も可)
一意性	条件次第で得やすい (凸性)	強モノトン性次第、一意性保証は難しいことも
利点	解釈・アルゴリズムの単純さ	一般性・柔軟性・拡張性

Beckmann型最適化問題の一般式

$$\min_{h \in \mathcal{H}} \sum_{a \in A} \int_0^{v_a(h)} t_a(s) ds$$

変分不等式 (VI) 形式のユーザー均衡の一般式

$$\langle C(h^*), h - h^* \rangle \geq 0 \quad \forall h \in H$$

h : 経路フロー、 v_a : リンク流量、 $\mathcal{H}$: 制約集合 (需要制約 + 非負制約)

研究の背景

ユーザー均衡の重要性と伝統的手法の限界

変分不等式(VI)

$$\langle C(h^*), h - h^* \rangle \geq 0 \quad \forall h \in H$$

経路集合 : P_{od}
 経路フロー : $h \in R^{|P|}$ (非負, 需要制約を満たす)
 経路コスト関数 : $C(h) \in R^{|P|}$ (フローに依存)
 可行(制約)集合 : $H = \{h \geq 0 \mid \Sigma^T h = q\}$
 R : 実数全体の集合
 h^* : 均衡時の経路フロー
 $\langle , \rangle$: 内積

1 : ワードロップの第一法則の数理的表現

2 : 可行集合の性質(解の存在保証)

3 : 射影法・オペレーター分割による実装

4 : 拡張性(選好: 経路コスト / 条件: 交通規制・個人異質性)

リンクベースVI: リンクフロー を変数とする形式

マルチクラスVI: 旅行者属性ごとに異なる選好を持たせる

動的VI : 時間に依存するフロー

ワードロップの第一法則

使用されている経路のコストは等しい

使用されていない経路のコストは同等 or 高い

誰も経路を変える動機がない状態 (ナッシュ均衡)

変分不等式

使用されている経路では

$$C_p(h^*) = \min_{r \in P_{od}} C_r(h^*) \quad \forall r \in P_{od}$$

使用されていない経路では

$$h_p^* = 0 \rightarrow C_p(h^*) \geq \min cost$$

$$\langle C(h^*), h - h^* \rangle \geq 0 \quad \forall h \in H$$

研究の背景

ユーザー均衡の重要性と伝統的手法の限界

変分不等式(VI)

$$\langle C(h^*), h - h^* \rangle \geq 0 \quad \forall h \in H$$

経路集合 : P_{od}
 経路フロー : $h \in R^{|P|}$ (非負, 需要制約を満たす)
 経路コスト関数 : $C(h) \in R^{|P|}$ (フローに依存)
 可行(制約)集合 : $H = \{h \geq 0 \mid \Sigma^T h = q\}$
 R : 実数全体の集合
 h^* : 均衡時の経路フロー
 $\langle , \rangle$: 内積

1 : ワードロップの第一法則の数理的表現

2 : 可行集合の性質(解の存在保証)

3 : 射影法・オペレーター分割による実装

4 : 拡張性(選好: 経路コスト / 条件: 交通規制・個人異質性)

リンクベースVI: リンクフローを変数とする形式

マルチクラスVI: 旅行者属性ごとに異なる選好を持たせる

動的VI : 時間に依存するフロー

2 : 可行集合の性質(解の存在保証)

VIの定義関数 $C(h^*)$ が連続かつ h が単調増加 → Brouwerの不動点定理により解の存在が保証

*可行集合の条件から、非負直交座標空間上で線形不等式制約を満たす連続性のある凸多面体になるため

解の存在は保証されているが、解の一意性は保証されていない(複数の均衡解が存在することもあり得る)

一意性の保証にはコスト関数の[強モノトン性]や[厳密凸性]などの追加性質が必要

研究の背景

ユーザー均衡の重要性と伝統的手法の限界

変分不等式(VI)

$$\langle C(h^*), h - h^* \rangle \geq 0 \quad \forall h \in H$$

経路集合 : P_{od}
 経路フロー : $h \in R^{|P|}$ (非負, 需要制約を満たす)
 経路コスト関数 : $C(h) \in R^{|P|}$ (フローに依存)
 可行(制約)集合 : $H = \{h \geq 0 \mid \Sigma^T h = q\}$
 R : 実数全体の集合
 h^* : 均衡時の経路フロー
 $\langle , \rangle$: 内積

1 : ワードロップの第一法則の数理的表現

2 : 可行集合の性質(解の存在保証)

3 : 射影法・オペレーター分割による実装

4 : 拡張性(選好: 経路コスト / 条件: 交通規制・個人異質性)

リンクベースVI: リンクフローを変数とする形式

マルチクラスVI: 旅行者属性ごとに異なる選好を持たせる

動的VI : 時間に依存するフロー

2 : 可行集合の性質(解の存在保証)

Browderの定理よりv非負制約・閉凸集合で少なくとも一つの解の存在が保証 > 解の一意性には次の性質の考慮が必要

性質	式	効果
モノトン性 (単調性)	$\langle C(h_1) - C(h_2), h_1 - h_2 \rangle \geq 0 \quad \forall h_1, h_2 \in H$	解の存在性と安定性が高まる
強モノトン性	$\langle C(h_1) - C(h_2), h_1 - h_2 \rangle > \mu \ h_1 - h_2\ ^2 \quad (\mu > 0)$	解の一意性
リプシッツ連続性	$\ C(h_1) - C(h_2)\ \leq L \ h_1 - h_2\ \quad (L > 0)$	コスト関数の滑らかさ・連続性を示し, 数値解法の収束速度の評価が可能に

研究の背景

可行集合の性質(解の存在保証)

2 : 解の存在保証証明 (簡略版-詳しくはAppendix C)

命題 :

VI形式で定義されたマルチクラスユーザー均衡
 $VI(C_\theta(h, x), H^{[m]})$ は、少なくとも1つの解を持つ

ステップ1 :

射影演算子の導入による問題の同値変形

射影 $P_H(z)$ を用いて固定点問題に変換(Appendix Cに基づく)

$$h^{*[m]} \in VI(C_\theta(h, x), H^{[m]}) \iff h^* = P_H(h^* - \alpha C_\theta(h^*, x))$$

ステップ2 :

$C_\theta(h, x)$ は定義より連続

可行集合は制約条件より、非負(非空), 閉空間, 凸関数なので

Browderの定理より、連続写像 $\times$ 閉凸集合 $\Rightarrow$ 固定点存在 $\Rightarrow$ 少なくとも1つの解の存在が保証

変分不等式(VI)

$$\langle C(h^*), h - h^* \rangle \geq 0 \quad \forall h \in H$$

経路集合	: P_{od}	
経路フロー	: $h \in R^{ P }$	(非負, 需要制約を満たす)
経路コスト関数	: $C(h) \in R^{ P }$	(フローに依存)
可行(制約)集合	: $H = \{h \geq 0 \mid \Sigma^T h = q\}$	
R	: 実数全体の集合	
h^*	: 均衡時の経路フロー	
$\langle \cdot, \cdot \rangle$	: 内積	

研究の背景

ユーザー均衡の重要性と伝統的手法の限界

変分不等式(VI)

$$\langle C(h^*), h - h^* \rangle \geq 0 \quad \forall h \in H$$

経路集合 : P_{od}
 経路フロー : $h \in R^{|P|}$ (非負, 需要制約を満たす)
 経路コスト関数 : $C(h) \in R^{|P|}$ (フローに依存)
 可行(制約)集合 : $H = \{h \geq 0 \mid \Sigma^T h = q\}$
 R : 実数全体の集合
 h^* : 均衡時の経路フロー
 $\langle , \rangle$: 内積

1 : ワードロップの第一法則の数理的表現

2 : 可行集合の性質(解の存在保証)

3 : 数値解法と適用条件による実装

4 : 拡張性(選好: 経路コスト / 条件: 交通規制・個人異質性)

リンクベースVI: リンクフローを変数とする形式

マルチクラスVI: 旅行者属性ごとに異なる選好を持たせる

動的VI : 時間に依存するフロー

非負制約 ($h \geq 0$) と 線形等式制約 ($\Sigma^T h = q$) を分割して プロジェクション法 (反復射影) で解ける

方法	特徴	適応対象	適応条件・特性
Frank-Wolfe法(線形化+方向探索)	射影不要, 凸最適化の繰り返し	経路ベース・凸VI	費用が連続・凸な場合
固定点反復法(Fixed-Point Iteration)	コスト関数が非線形でも可能	モノトンVI	反復式の定義・モノトン性がある場合
Broyden法(準ニュートン法)	勾配不明でも適用可能	非線形VI	非線形かつ非凸でも適応できるが収束保証が弱い
オペレーター分割法 (Operator Splitting / Forward-Backward)	並列計算・拡張性あり	大規模VI・モノトン演算子和	ネットワーク全体・時間依存構造に有利

研究の背景

ユーザー均衡の重要性と伝統的手法の限界

変分不等式(VI)

$$\langle C(h^*), h - h^* \rangle \geq 0 \quad \forall h \in H$$

経路集合 : P_{od}
 経路フロー : $h \in R^{|P|}$ (非負, 需要制約を満たす)
 経路コスト関数 : $C(h) \in R^{|P|}$ (フローに依存)
 可行(制約)集合 : $H = \{h \geq 0 \mid \Sigma^T h = q\}$
 R : 実数全体の集合
 h^* : 均衡時の経路フロー
 $\langle , \rangle$: 内積

1 : ワードロップの第一法則の数理的表現

2 : 可行集合の性質(解の存在保証)

3 : 射影法・オペレーター分割による実装

4 : 拡張性(選好: 経路コスト / 条件: 交通規制・個人異質性)

リンクベースVI: リンクフローを変数とする形式

マルチクラスVI: 旅行者属性ごとに異なる選好を持たせる

動的VI : 時間に依存するフロー

VIの枠組みは

選好(経路コスト)や条件(交通制約/エージェント異質性)を含めた拡張が可能に(制約条件の変更によって)

拡張の方向	数式	構造的特徴
マルチクラスVI	$\langle C^k(h^{k*}), h^k - h^{k*} \rangle \geq 0 \quad \forall h^k \in H^k, \forall k \in K$	各クラス k (例: 自動車・バス・歩行者)ごとにコスト関数 C^k などを定義. 全体として複数のVIが連立する「マルチクラス均衡」構造
動的VI	$\langle C_t(h_t^*), h_t - h_t^* \rangle \geq 0. \quad \forall h_t \in H_t, \forall t \in T$	時間 t に依存するコスト C_t , フロー h_t を, 全時間区間 T で連続的または離散的にVIを定義. 動的ユーザー均衡 (DUE) モデルと対応
リンクベースVI	$\langle c_a(h^*), h_a - h_a^* \rangle \geq 0 \quad \forall h_a \in H_a, \forall a \in A$	各リンク(道路区間) a に対するリンクコスト c_a とリンクフロー h_a をリンク単位で定義. ネットワークの局所特性を表現.

3 : Methodology

モデルの全体像

end to endとImplicit neural network の概念

学習データから経路選択・ネットワーク均衡まで直接推定する(End to end型)

入力：OD需要 $q^{[m]}$ コンテキスト情報 x

OD需要を入力する意味

可行(制約)集合の生成 : $H = \{h \geq 0 \mid \Sigma^T h = q\}$

コンテキスト情報とは

- Traveler characteristics(旅行者情報)
- Road network features (道路情報)

Implicit Layer : 変分不等式(VI) を解く

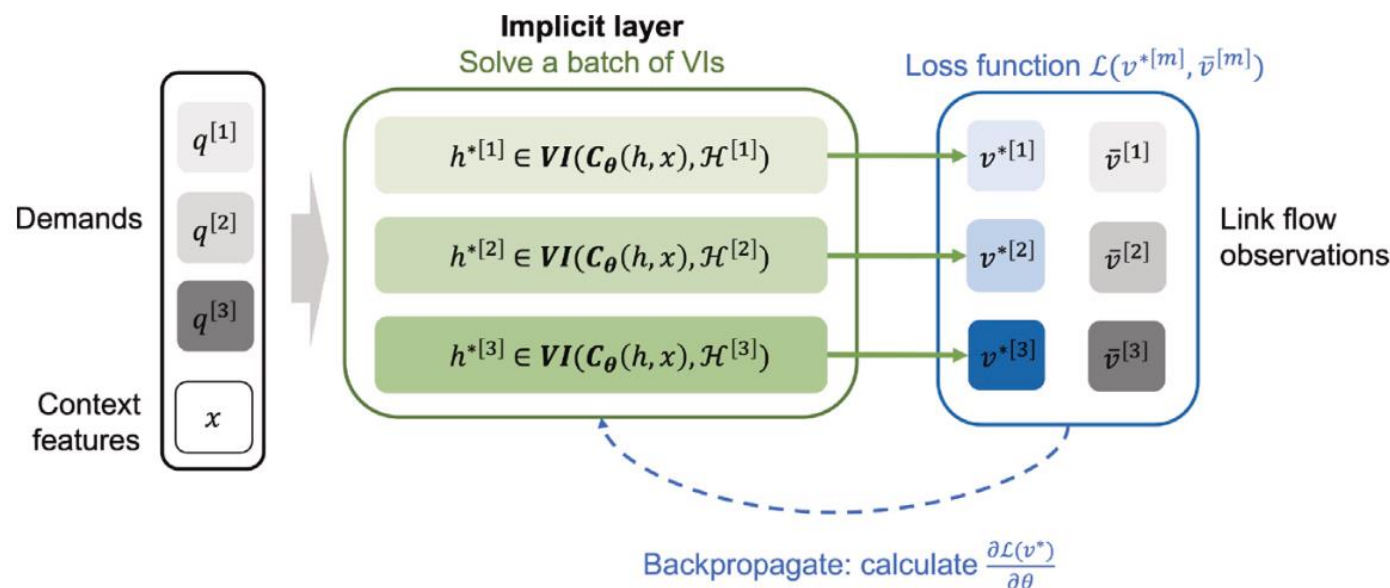
OD需要の制約を満たす条件のもと変分不等式を解き h を更新し、均衡解を求める

$$h^{*[m]} \in VI(C_\theta(h, x), H^{[m]})$$

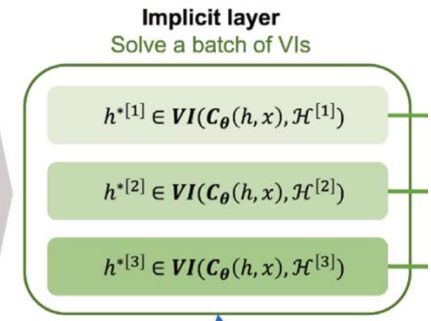
損失関数の計算 :

均衡解 $h^{*[m]}$ から計算した

リンク流量 $v^{*[m]}$ と観測流量 $\bar{v}^{[m]}$ を比較



3.1 : Neural-network-based VI formulation of UE



Implicit Layerの中身

Neural-network-based VI formulation of

学習データから経路選択・ネットワーク均衡まで直接推定する(end to end型)

Weight Net F_θ : 個人の選好に応じたコスト重み付を学習するNN

入力値 : x^R

- ・ 旅行者属性 : 収入, 時間価値, 年齢層

出力値 : w

- ・ 経路の重みベクトル

Attribute Net G_θ : リンク状態(混雑など)に応じた経路属性の出力するNN

入力値

- ・ 道路ネットワーク特性 : x^G
自由流速度, リンク容量, リンク長, 道路種別
- ・ 経路フロー : h

出力値

- ・ 経路の属性ベクトル : $\pi = G_\theta(h^{[m]}, x^G)$

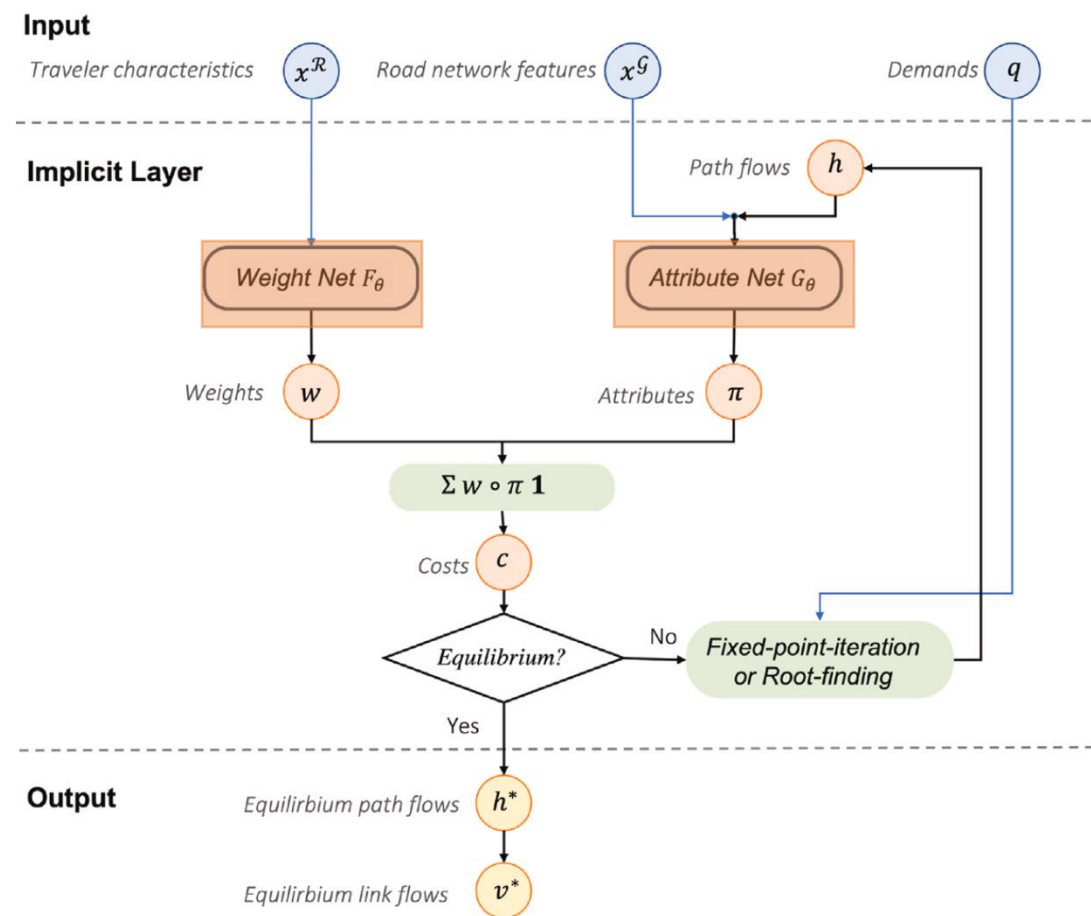


Fig. 2. Illustration of the end-to-end learning framework. The superscripts for a sample m are omitted.

Attribute Net

道路情報と更新した経路フローによる道路属性 π の算出

具体的な道路特徴の学習方法

Link Block g_{θ}^A : リンク属性 π_a^A を出力

Node Block g_{θ}^N : ノード属性 π_n^N を出力

Path Block g_{θ}^P : パス属性 π_p^P を出力

各特徴の次元をPath 行列の次元に揃える

$$\pi = [\Lambda \pi^A, \Gamma \pi^N, \pi^P]$$

Λ : 経路-リンク対応行列 (path-link incidence matrix)

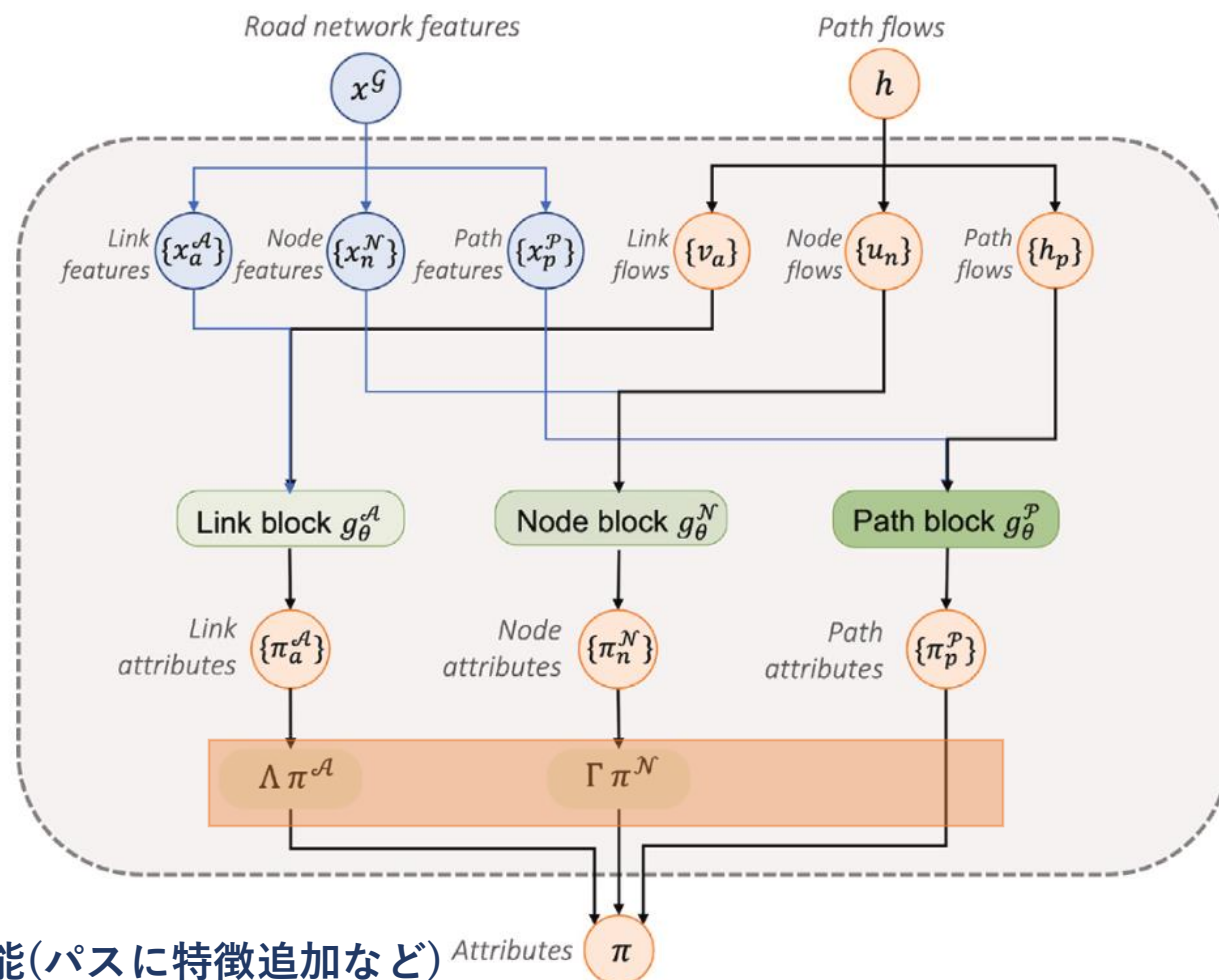
Γ : 経路-ノード対応行列 (path-node incidence matrix)

三つに分解する意味

トポロジー非依存性 : 部分的なネットワークサイズ変更が可能(パスに特徴追加など)

学習の効率化 : 局所単位で処理でき、パラメータ数を抑制

表現力 : マルチスケールで状態を表現できる



Weight Net

個人属性による経路選好(重み)を学習する

ODペア毎の異なる旅行者特性を考慮して経路選択時の重みを学習する

ODペアに対する重み w

$$w = F_{\theta}(x^R) = \{f_{\theta}(x_r^R)^{\top}\}_{r \in R}$$

旅行者の選好を個人属性・社会属性から重み付け

* 同じODペアの旅行者はすべて同じ選考を共有する単一クラス

注意点

- ・ f_{θ} のパラメータはすべてのODペアで共有されている
 - ・ 個人属性はOD毎に
 - ・ 連続値の場合 >> 平均値
 - ・ バイナリー値 >> 最頻値
- を用いて算出

モデルの拡張性

解釈可能なニューラルネットワーク型離散選択モデルを組み込むことで特に行動の解釈性が必要な場面において有用となる と言及

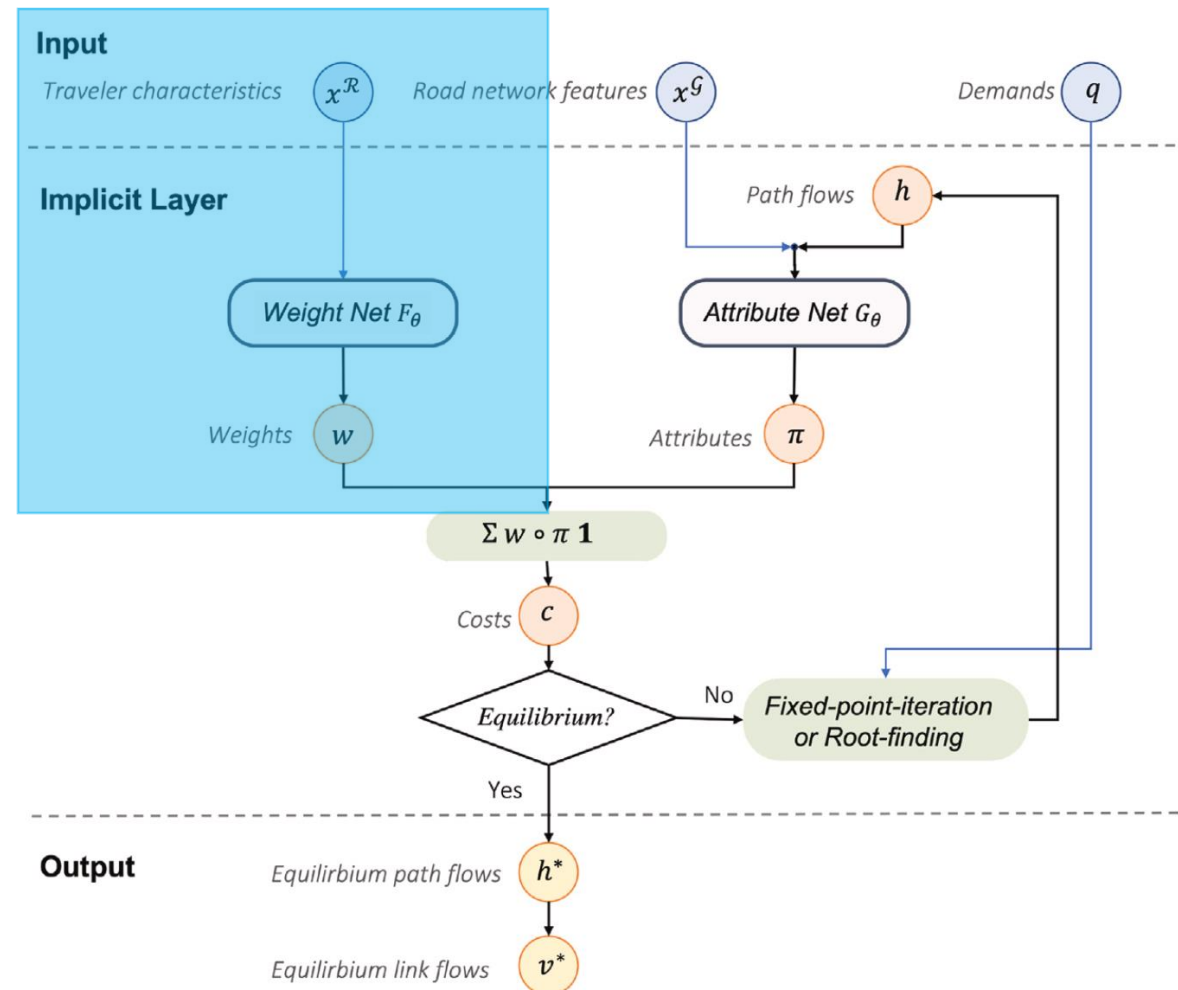


Fig. 2. Illustration of the end-to-end learning framework. The superscripts for a sample m are omitted.

Costs function

学習した特徴量とアダマール積を用いたコスト計算

学習データから経路選択・ネットワーク均衡まで直接推定する(end to end型)

経路コスト c : 現在の経路フローに基づいたコストを計算

$$c = \sum (w \circ \pi \circ \mathbf{1})$$

$\mathbf{1}$: Hadamard(アダマール)積の数式的な整合性のためのインジケータベクトル

w : 個人属性毎に計算された経路の重みベクトル

π : $h^{[m]}$ (更新した経路フロー情報)と道路特徴で計算された経路属性ベクトル

つまり、経路コスト c を [旅行者属性 w] * [道路属性 π] の線形和で表現

本研究では....

同一OD間の個人を同じ選好を共有する単一クラスと仮定

OD毎に複数のサブクラスを作れば、マルチクラスVIで実装も可能

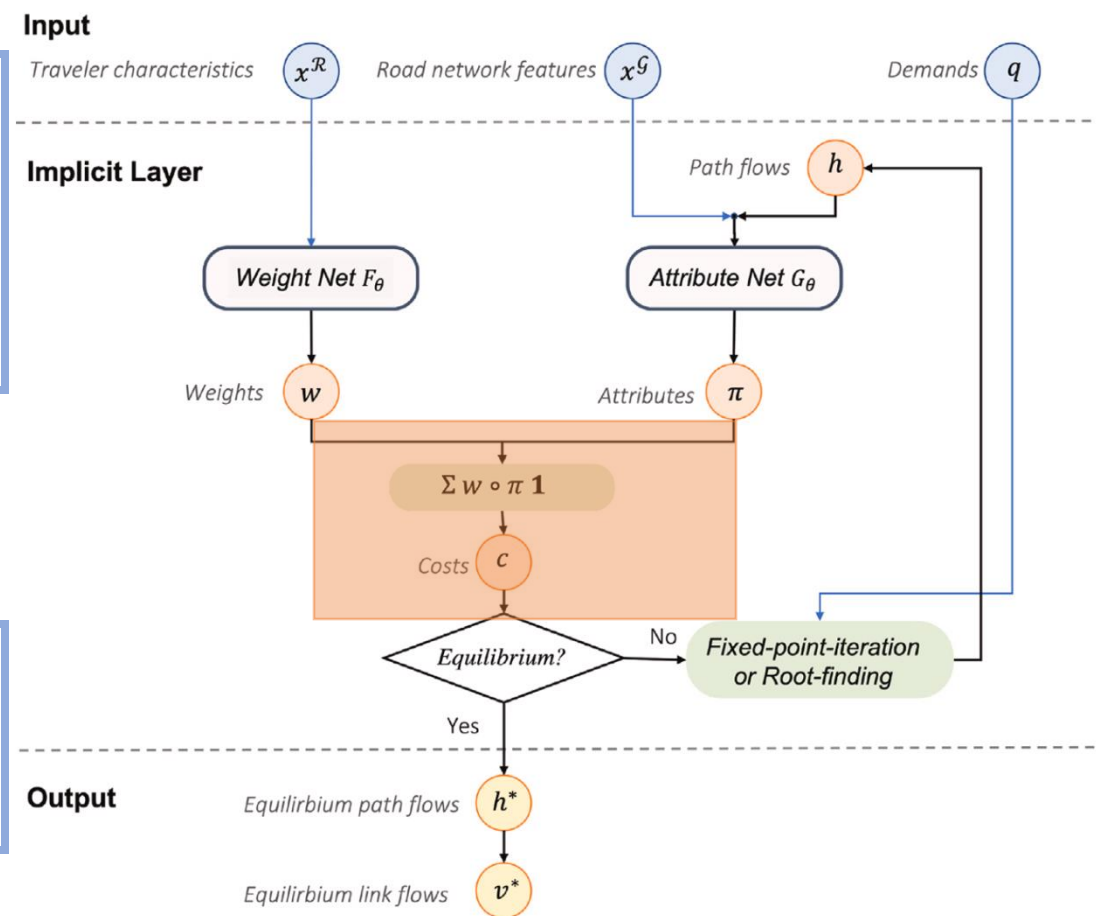


Fig. 2. Illustration of the end-to-end learning framework. The superscripts for a sample m are omitted.

Regularization of cost function

変分不等式を解くための重みに対する制約条件

変分不等式の解を持たせるためにNNの正則化を用いて **モノトン性 & リプシッツ連続性** を持たせる

(簡略版-詳しくはAppendix B)

活性化関数にSoft Plus, ReLU 関数を用いる >> **モノトン性・リプシッツ連続**を持つ (部分的な活性化関数自体は)

ネットワーク全体として**モノトン**かつ**リプシッツ連続**でないといけない

モノトン性

1 : Attribute Blocks $g_{\theta}^A, g_{\theta}^N, g_{\theta}^L$ の制約

- ・列方向に単調になるように

2 : Attribute Netにおける活性化関数の選択

$$y^{(l)} = \sigma^{(l)} \left(W^{(l)} y^{(l-1)} + b^{(l)} \right)$$

$\sigma^{(l)}$: 活性化関数
 $y^{(l-1)}$: 第 l 層の入力

$\sigma^{(l)} = \text{ReLU}$ を用いてすべての線形層の重みを正に

3 : Weight Netの最終線形層の重み制約

重み行列 w の要素をすべて正に

強モノトン性

1 : Attribute Netにおける活性化関数の選択

$$y^{(l)} = \sigma^{(l)} \left(W^{(l)} y^{(l-1)} + b^{(l)} \right)$$

$\sigma^{(l)}$: 活性化関数
 $y^{(l-1)}$: 第 l 層の入力

2 : Attribute Netの線形層の重み制約

重み行列 $W^{(l)}$ が正定値かつ対称になるように制約

$$W^{(l)} = B^{\top} B + \iota I$$

$B^{\top} B$: 対称行列かつ半正定値行列で固有値 ≥ 0
(B は任意の実行列)
 ιI : 正定値で固有値は $\iota > 0$

3 : 正定値行列の線形写像は常に強モノトン

$$\langle W(x - y), x - y \rangle = (x - y)^{\top} W (x - y) > 0$$

Regularization of cost function

変分不等式を解くための重みに対する制約条件

変分不等式の解を持たせるためにNNの正則化を用いて **モノトン性 & リプシッツ連続性** を持たせる
(簡略版-詳しくはAppendix B)

活性化関数にSoft Plus, ReLU 関数を用いる >> **モノトン性・リプシッツ連続** (部分的な活性化関数自体は)
ネットワーク全体としてモノトンかつリプシッツ連続でないといけない

リプシッツ連続性 定義式: $\|C(h_1) - C(h_2)\| \leq L \|h_1 - h_2\|$

Attribute Netの線形層のヤコビ行列を正規化(スペクトル正規化)

スペクトルノルム(2-ノルム)の定義

$$\|W\|_2 = \sup_{\|x\|_2=1} \|Wx\|_2$$

入力ベクトルを2ノルムで1に
正規化した際、出力ベクトル
がどれだけ伸びるかの最大値

||

最大特異値

$$\|W\|_2 = \sigma_{\max}(W)$$

$f(x) = Wx$ は線形写像なので定義式より

$$\|Wx - Wy\| = \|W(x - y)\| \leq \|W\|_2 \cdot \|x - y\| \Rightarrow L = \|W\|_2 \quad -(1)$$

NNは下記の活性化関数の合成関数なので

$$y^{(l)} = \sigma^{(l)} \left(W^{(l)} y^{(l-1)} + b^{(l)} \right)$$

まとめると
V

$$f(x) = \sigma^{(L)}(W^{(L)} \dots \sigma^{(1)}(W^{(1)}x + b^{(1)}) \dots + b^{(L)}) \quad -(2)$$

式(1), (2)より以下の不等式が成り立つ

$$L_{\text{total}} \leq \prod_{l=1}^L \|W^{(l)}\|_2 \cdot \|\sigma^{(l)}\|_{\text{Lip}}$$

$\|W^{(l)}\|_2$: 第 l 層の重み行列のスペクトルノルム (最大特異値)

$\|\sigma^{(l)}\|_{\text{Lip}}$: 活性化関数のリプシッツ定数 (例: ReLU は 1、SoftPlus も 1)

活性化関数がReLUやSoftPlusのようにリプシッツ連続であればNN(活性化関数を含む線形層の合成関数)も、リプシッツ連続である

これはリプシッツ定数の上界なので発散を防ぐために最大特異値を用いて 1 or それ以下に正規化

$$\bar{W}^{(l)} = \frac{W^{(l)}}{\sigma_{\max}(W^{(l)})} \quad \text{または} \quad \bar{W}^{(l)} = \frac{W^{(l)}}{\max(\sigma_{\max}(W^{(l)}), \epsilon)}$$

3.2 : End-to-end learning with an implicit layer

Implicit layer 内の学習

変分不等式による制約とNNによる均衡解の導出

MPEC (均衡制約付き数理計画問題)で定式化

$$\min_{\theta} \mathbb{E}_{q \sim \mathcal{D}} [\mathcal{L}(v^{*[m]}, v^{[m]})] \quad \text{s.t. } h^{*[m]} \in \text{VI}(C_{\theta}(h, x), \mathcal{H}^{[m]}), \quad v^{*[m]} = \Lambda^{\top} h^{*[m]}$$

学習用データセット $\{x^{[m]}, q^{[m]}, v^{[m]}\}_{m=1}^M$

最小化する目的関数：損失関数(均衡解と観測値の)
制約条件：Wardrop 均衡 (VIで表現)

射影によって変分不等式を補助固定点問題に変形

$$h^* = P_{\mathcal{H}}(h^* - \alpha C(h^*))$$

現在の射影 $P_{\mathcal{H}}$ を可行集合 H 下で閉形式で計算するのは困難
オペレータ分割法を用いて、2つの制約に段階的に射影する

可行集合の分割による2段階の射影

可行(制約)集合： $H = \{h \geq 0 \mid \Sigma^{\top} h = q\}$

変分不等式の可行(制約)集合は以下の二つ
集合を各部分集合へと分解

1： $\mathcal{H}_1 = \{h \in \mathbb{R}^{|\mathcal{P}|} \mid h \succeq 0\}$ ：非負制約

2： $\mathcal{H}_2^{[m]} = \{h \mid \Sigma^{\top} h = q^{[m]}\}$ ：OD需要制約 (等式制約)

1： $\hat{h}^{(k)} = P_{\mathcal{H}_1}(z^{(k)})$

2： $z^{(k+1)} = T_{\theta}(z^{(k)}, x)$

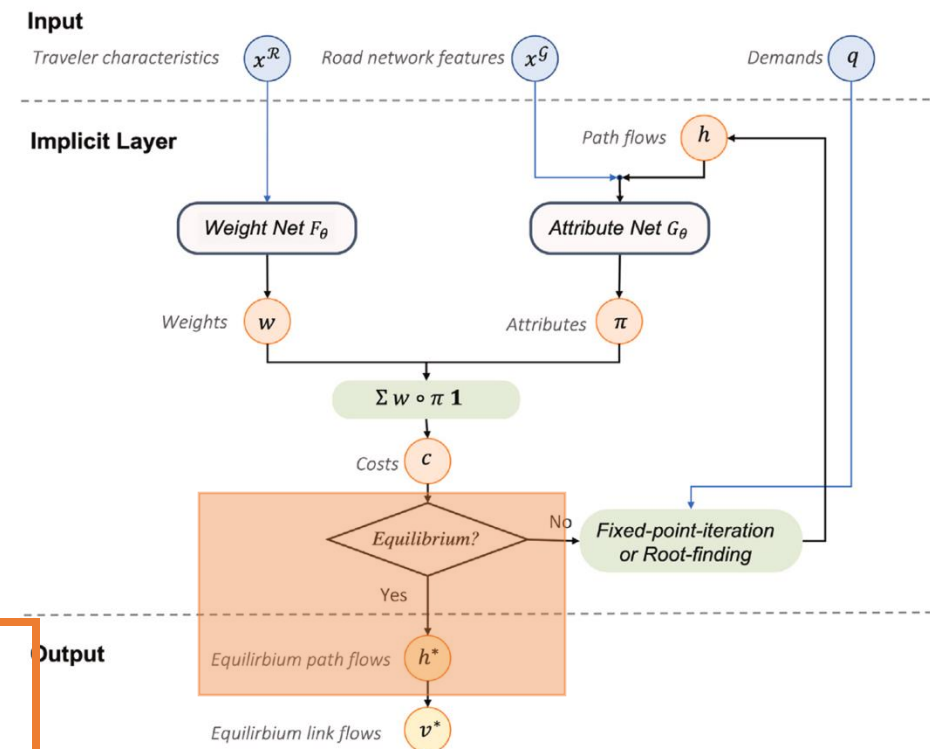
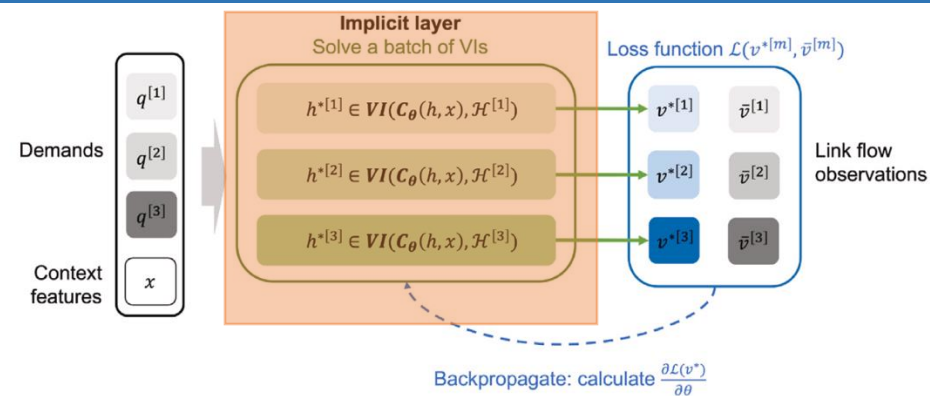


Fig. 2. Illustration of the end-to-end learning framework. The superscripts for a sample m are omitted.

Implicit layer 内の学習

変分不等式による制約とNNによる均衡解の導出

算出された均衡パスフローと観測値の損失関数の最小化

- 1 : $\hat{h}^{(k)} = P_{\mathcal{H}_1}(z^{(k)})$
 - 2 : $z^{(k+1)} = T_{\theta}(z^{(k)}, x)$
- *標準固定点反復法の場合は以下の定式化
- $$z^{(k+1)} = z^{(k)} - \alpha (\hat{h}^{(k)} - P_{\mathcal{H}_2}(\hat{h}^{(k)} - \alpha C_{\theta}(\hat{h}^{(k)}, x)))$$

補助固定点問題への射影1, 2により、経路フローの均衡解 h^* (一意性はない) が求まる
解き方は様々 (詳細: Appendix)

- 1 : 標準固定点反復法 反復構造で、簡易かつ安定だが収束は遅め
- 2 : Anderson Mixing 過去の反復を最適に合成して高速収束を狙う加速法
- 3 : Weighted Ergodic Iteration 収束率を理論的に改善する重み付き平均化法
- 4 : Root finding(Broyden法) 勾配がなくても方向を近似する準ニュートン法で高精度解法

損失関数のバックプロパゲーション

$$\min_{\theta} \mathbb{E}_{q \sim \mathcal{D}} [\mathcal{L}(v^{*[m]}, v^{[m]})] \quad \text{s.t. } h^{*[m]} \in \text{VI}(C_{\theta}(h, x), \mathcal{H}^{[m]}), \quad v^{*[m]} = \Lambda^{\top} h^{*[m]}$$

損失関数 $\mathcal{L}(v^{*[m]}, v^{[m]})$ の勾配は補助固定点問題の固定点解 z^* の微分が必要

補助固定点問題の試行回数を逐次逆伝播するのは計算コストが高い
沈黙関数定理(implicit Function Theorem)を用いると z^* を
NNのパラメータ θ の連続関数と見なすことができる

$$\frac{\partial z^*}{\partial \theta} = \left(I - \frac{\partial T_{\theta}(z^*, x)}{\partial z^*} \right)^{-1} \cdot \frac{\partial T_{\theta}(z^*, x)}{\partial \theta}$$

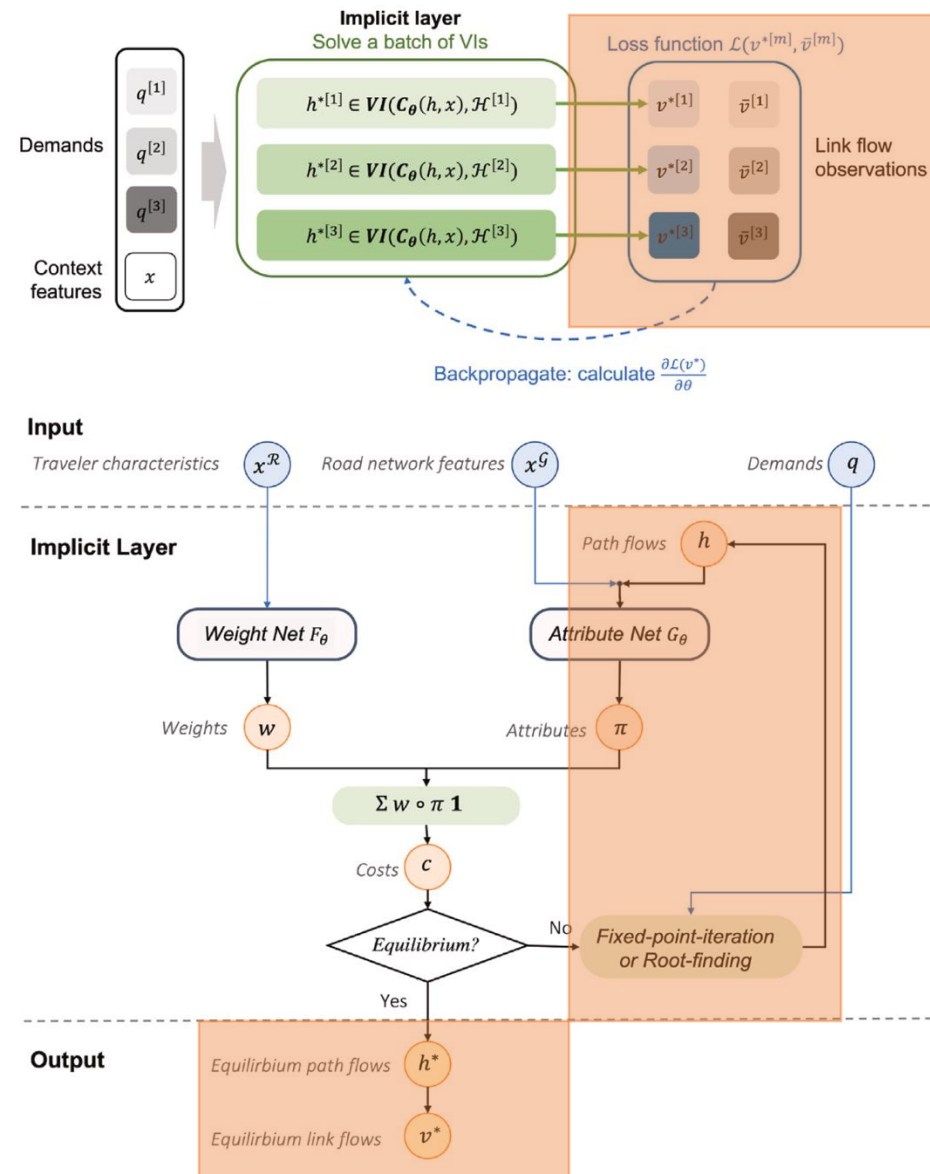


Fig. 2. Illustration of the end-to-end learning framework. The superscripts for a sample m are omitted.

Implicit layer 内の学習

変分不等式による制約とNNによる均衡解の導出

損失関数のバックプロパゲーション

沈黙関数定理(implicit Function Theorem)

z^* をNNのパラメータ θ の連続関数と見なせる

$$\frac{\partial z^*}{\partial \theta} = \left(I - \frac{\partial T_{\theta}(z^*, x)}{\partial z^*} \right)^{-1} \cdot \frac{\partial T_{\theta}(z^*, x)}{\partial \theta}$$

上の関数のうち、逆行列の計算が重い

“Jacobian-free Backpropagation”によって線形固定点問題に帰着できる

$$g = \left(\frac{\partial T_{\theta}(z^*, x)}{\partial z^*} \right)^{\top} g + \frac{\partial \mathcal{L}}{\partial z^*}$$

この式の収束条件は ヤコビアン の最大固有値の絶対値が1未満であること

リプシッツ連続性の上界で抑えるためにスペクトル正規化をここでも行うことで要件を満たす

$$\|T_{\theta}(z_1) - T_{\theta}(z_2)\| \leq L \cdot \|z_1 - z_2\| \quad \forall z_1, z_2$$

最終的な写像のLipschitz 定数 $L < 1$ を満たすように活性化関数を設計

$$\frac{\partial \mathcal{L}}{\partial \theta} = \left(\frac{\partial T_{\theta}}{\partial \theta} \right)^{\top} g$$

これにより、損失関数を得ることができる

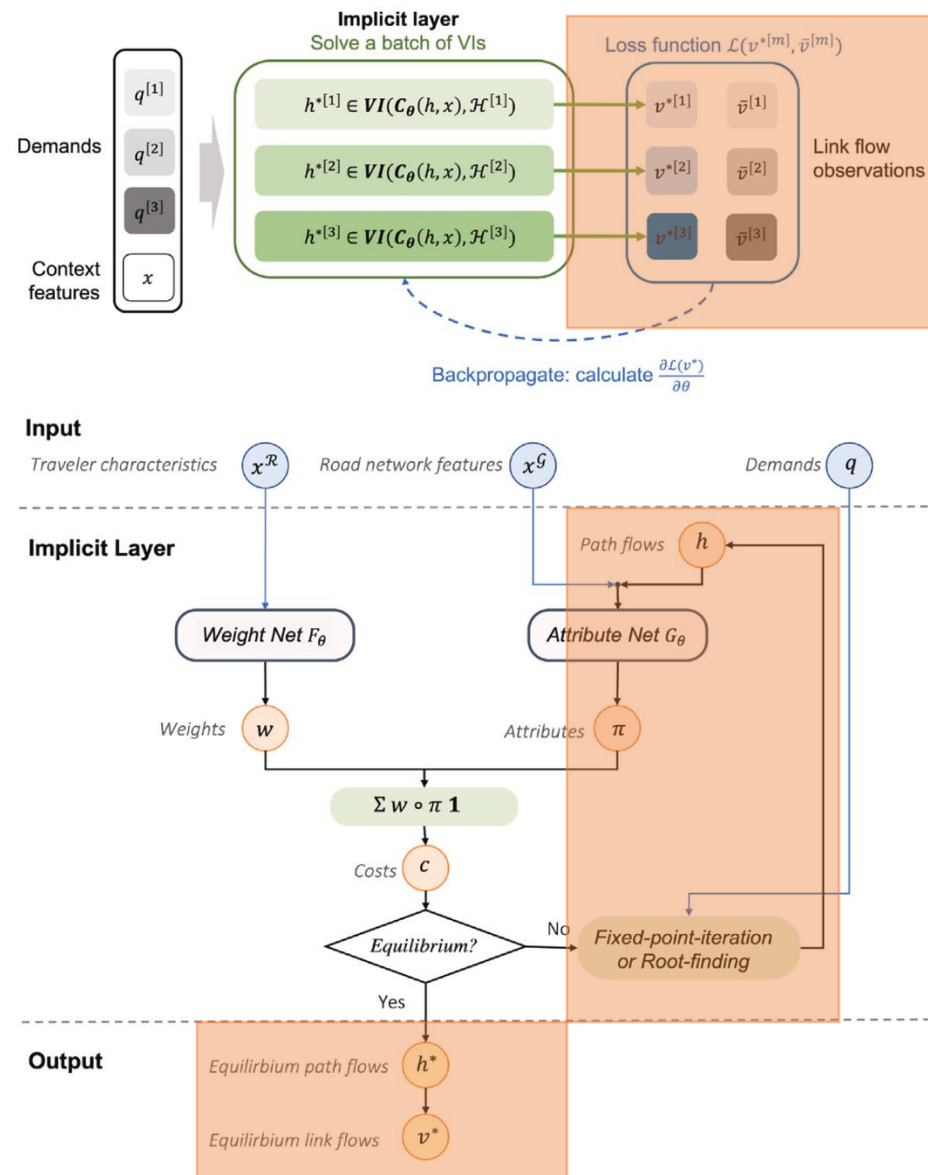


Fig. 2. Illustration of the end-to-end learning framework. The superscripts for a sample m are omitted.

4 : Experiments

検証前の準備

使用ネットワーク：Sioux Fallsネットワーク（リンク数76、ノード数28、ODペア数528）

各ODペアで自由流状態での移動時間が短い上位3経路を選択, 経路集合 P_r とする

検証シナリオ

1：訓練データの設定

1.1：特徴量の設定

1.2：用いる関数の設定

2：ニューラルネットの構造の設定

2.1：Attribute Net (Link, Node, Path Block)

2.2：Weight Net

2.3：学習の計算手段

3：性能評価方法の定義

3.1：アルゴリズムの組み合わせによる比較

3.2：MAPE(平均絶対誤差率)での評価

実験結果：予測精度の比較・ロバストの評価

1：従来手法との比較

1.1：DUE, SUE, SUE-2 との比較

2：使用アルゴリズム間の比較

2.1：順伝播のアルゴリズムの収束速度

2.2：アルゴリズム間のMAPE比較

2.3：逆伝播のアルゴリズムの比較

2.4：スペクトル正規化による計算速度上昇評価

3：計算のロバスト性

3.1：センサー設置の欠落

3.2：OD需要のノイズ(誤推定)

3.3：経路集合の不完全性

検証シナリオ

使用ネットワーク：Sioux Fallsネットワーク（リンク数76、ノード数28、ODペア数528）
各ODペアで自由流状態での移動時間が短い上位3経路を選択, 経路集合 P_r とする

1：訓練データの生成

1.1：特徴量の設定

各ODペアに個人特徴量を付与

- 1：連続値の特徴量 x_{1r} = 所得
- 2：バイナリ特徴量 x_{2r} = 旅行目的（買い物=1, その他=0）

コストの定義

- 1：リンク所要時間 $t_a(v_a)$
- 2：交差点(Node)遅延 $d_n(u_n)$
- 3：旅行不快感 e_p
- 4：総経路コスト c_p

1.2：用いるコスト関数の設定

- 1：リンク所要時間 >> リンクコスト関数(BPR関数)
- 2：交差点遅延モデル >> Jaihani's 指数関数モデル
- 3：旅行不快関数 >> 悪リンク割合 指数関数モデル
- 4：総経路コスト関数 >> 1~3の総和に重みを掛けた式

リンクコスト関数(BPR関数)

$$t_a(v_a) = t_a \left(1 + 0.15 \left(\frac{v_a}{c_a} \right)^4 \right)$$

t_a ：リンク a の自由流時の所要時間

v_a ：リンク a の交通量

c_a ：リンク a の容量

Jaihani's 指数関数モデル

$$d_n(u_n) = d_n \left(\frac{u_n}{c_n} \right)^{\beta_n} + \gamma_n$$

u_n ：ノード n に流入する総交通量（周辺リンクの合計）

$d_n, c_n, \beta_n, \gamma_n$ ：交差点の設計に依存するパラメータ

悪リンク割合 指数関数モデル

$$e_p = \exp(\alpha x_p) + \beta$$

$$x_p = \frac{\text{悪リンク長}}{\text{全経路長}}, \quad 0 \leq x_p \leq 1$$

$\alpha = 2, \beta = -1$ と設定

すべてのリンクが良好であれば $e_p = 0$ となる。

1~3の総和に重みを掛けた(アダマール積)

$$c_p = \sum_{a \in p} t_a(v_a) + w_r^d \sum_{n \in p} d_n(u_n) + w_r^e e_p$$

$$w_r^d = (5.5 + x_{1r}) \cdot \exp(1 - x_{2r})$$

$$w_r^e = 10 \cdot x_{1r} \cdot x_{2r}$$

w_r^d ：ノード遅延に対するODペアごとの重み

w_r^e ：不快感に対するODペアごとの重み

検証シナリオ

使用ネットワーク：Sioux Fallsネットワーク（リンク数76、ノード数28、ODペア数528）
各ODペアで自由流状態での移動時間が短い上位3経路を選択, 経路集合 P_r とする

2：ニューラルネットの構造設定

2.1：Attribute Net (Link, Node, Path Block)

Link Block

事前にキャリブレーションされたBPR関数

Node・Path Block

3層の全結合層 (fully connector layer)

各層：4個のニューロン

活性化関数：LeakyReLU 関数

2.2：Weight Net

3層の全結合層 (fully connector layer)

各層：4個のニューロン

活性化関数：LeakyReLU 関数

2.3：学習の計算手段

順伝播(forward)：

F：固定点反復法

FA：Anderson Mixing

FW：Weighted Ergodic Iteration

R：Broyden法

F-R：F + R

FA-R：FA + R

FW-R：FW + R

逆伝播(backward)：

JF：Jacobian-free Backpropagation

NA：Newman Approximation

FA：Fixed point Approximation

ブロック	入力	出力	NNによる学習
Link block	フロー v_a 、特徴 x_a	移動時間 (BPR)	本実験では固定式 (学習なし)
Node block	ノード流 u_n 、交差点特徴	遅延時間	学習対象
Path block	悪条件リンクの割合 x_p	不快感	学習対象
Weight Net	所得・目的地の属性	重み w_d, w_e	学習対象

検証シナリオ

3：性能評価方法

3.1：アルゴリズムの組み合わせ

Table 1
Forward algorithm hyperparameters.

Name	κ_1	α_1	β_1	γ_1	κ_2	α_2	β_2	γ_2
F	1e3	1e-3	1	0.9	–	–	–	–
FA	1e3	1e-3	1	0.9	–	–	–	–
FW	1e3	1e-3	1	0.9	–	–	–	–
R	–	–	–	–	1e2	1e-3	1	0.9
F-R	1e3	1e-3	1	0.9	1e2	1e-3	1	0.9
FA-R	10	1e-3	1	0.9	10	1e-3	1	0.9
FW-R	1e3	1e-3	1	0.9	1e2	1e-3	1	0.9

3.2：MAPE(平均絶対誤差率)での評価

リンクフロー予測のパーセンテージ誤差

$$\eta = \frac{1}{|\mathcal{A}|} \sum_{a \in \mathcal{A}} \left| \frac{v_a^* - v_a}{v_a} \right| \times 100\%$$

$\mathcal{A}$ ：ネットワーク内の全リンク集合

v_a^* ：リンク a における予測フロー

v_a ：リンク a における真値 (ground-truth)

アルゴリズム名	内容・特徴
F	固定点反復法 (Fixed Point Iteration) だけを使用。シンプルだが収束速度は遅め。
FA	Anderson Mixing を加えた固定点反復。過去の情報を加重平均して収束を高速化。
FW	Weighted Ergodic Iteration (加重エルゴード反復法)。対称 Jacobian 条件下で収束性を向上。
R	Root-finding (Broyden 法) のみ使用。高速だが不安定なケースあり。
F-R	固定点反復で大まかに収束後、Root-finding に切り替え。
FA-R	Anderson Mixing → Broyden 法のハイブリッド (ただし最大反復数が10と小さい)。
FW-R	Weighted Ergodic → Broyden 法のハイブリッド (最も堅牢・高精度とされる)。
記号	意味
κ_1	固定点反復 (Fixed Point Iteration) 系アルゴリズムにおける最大反復回数
α_1	固定点反復における初期ステップサイズ (学習率に相当)
β_1	相対残差が減少しなかった場合の閾値 (1.0なら「悪化しなければ」)
γ_1	ステップサイズ調整係数 (収束しなければ $\alpha_1 \times \gamma_1$ で更新)
κ_2	Root-finding (準ニュートン法: Broyden 法) での最大反復回数
α_2	Root-finding の初期ステップサイズ
β_2	Root-finding における収束性の閾値
γ_2	Root-finding のステップサイズ更新係数

検証シナリオ

3：性能評価方法

3.1：アルゴリズムの組み合わせ

Table 1
Forward algorithm hyperparameters.

Name	κ_1	α_1	β_1	γ_1	κ_2	α_2	β_2	γ_2
F	1e3	1e-3	1	0.9	—	—	—	—
FA	1e3	1e-3	1	0.9	—	—	—	—
FW	1e3	1e-3	1	0.9	—	—	—	—
R	—	—	—	—	1e2	1e-3	1	0.9
F-R	1e3	1e-3	1	0.9	1e2	1e-3	1	0.9
FA-R	10	1e-3	1	0.9	10	1e-3	1	0.9
FW-R	1e3	1e-3	1	0.9	1e2	1e-3	1	0.9

3.2：MAPE(平均絶対誤差率)での評価

リンクフロー予測のパーセンテージ誤差

$$\eta = \frac{1}{|\mathcal{A}|} \sum_{a \in \mathcal{A}} \left| \frac{v_a^* - v_a}{v_a} \right| \times 100\%$$

$\mathcal{A}$: ネットワーク内の全リンク集合

v_a^* : リンク a における予測フロー

v_a : リンク a における真値 (ground-truth)

アルゴリズム名	内容・特徴
F	固定点反復法 (Fixed Point Iteration) だけを使用。シンプルだが収束速度は遅め。
FA	Anderson Mixing を加えた固定点反復。過去の情報を加重平均して収束を高速化。
FW	Weighted Ergodic Iteration (加重エルゴード反復法)。対称 Jacobian 条件下で収束性を向上。
R	Root-finding (Broyden 法) のみ使用。高速だが不安定なケースあり。
F-R	固定点反復で大まかに収束後、Root-finding に切り替え。
FA-R	Anderson Mixing → Broyden 法のハイブリッド (ただし最大反復数が10と小さい)。
FW-R	Weighted Ergodic → Broyden 法のハイブリッド (最も堅牢・高精度とされる)。
記号	意味
κ_1	固定点反復 (Fixed Point Iteration) 系アルゴリズムにおける最大反復回数
α_1	固定点反復における初期ステップサイズ (学習率に相当)
β_1	相対残差が減少しなかった場合の閾値 (1.0なら「悪化しなければ」)
γ_1	ステップサイズ調整係数 (収束しなければ $\alpha_1 \times \gamma_1$ で更新)
κ_2	Root-finding (準ニュートン法: Broyden 法) での最大反復回数
α_2	Root-finding の初期ステップサイズ
β_2	Root-finding における収束性の閾値
γ_2	Root-finding のステップサイズ更新係数

5 : Result

比較結果

結果の比較：予測精度の比較

1：従来手法との比較

1.1：DUE, SUE, SUE-2 との比較

1.1.2：モデルの性能 (In-distribution test)

1.1.3：モデルの汎化性能(Out-distribution test)

Table 2
MAPE of different network equilibrium models. MAPEs are shown in percentage.

In-distribution test					
Demand	Capacity	DUE η_0	SUE $\Delta\eta$	SUE-2 $\Delta\eta$	Implicit $\Delta\eta$
Base	Default	20.6	-4.7	-11.8	-15.0
Uncongested	Default	12.5	-3.1	-0.2	-3.4
Congested	Default	13.41	-0.6	-4.4	-10.2
	Mean	15.5	-2.8 (-18.2%)	-5.4 (-35.1%)	-9.5 (-61.5%)
Out-of-distribution test					
Demand	Capacity	DUE η_0	SUE $\Delta\eta$	SUE-2 $\Delta\eta$	Implicit $\Delta\eta$
Base	Default	22.3	-7.3	-14.4	-16.6
	-50%	11.3	+13.4	-1.6	-7.9
	+50%	8.1	+4.8	-1.0	-1.3
Uncongested	Default	23.4	-8.5	-15.6	-14.9
	-50%	12.1	+12.6	-2.4	-4.1
	+50%	10.4	+2.5	-3.3	-1.1
Congested	Default	13.8	-3.5	-6.3	-10.1
	+50%	11.9	-3.5	-5.2	-6.4
	Mean	14.2	+1.3 (+9.2%)	-6.2 (-44.0%)	-7.8 (-55.1%)

2：使用アルゴリズム間の比較

2.1：順伝播のアルゴリズムの収束速度

順伝播(forward)：

F：固定点反復法

FA：Anderson Mixing

FW：Weighted Ergodic Iteration

R：Broyden法

F-R：F + R

FA-R：FA + R

FW-R：FW + R

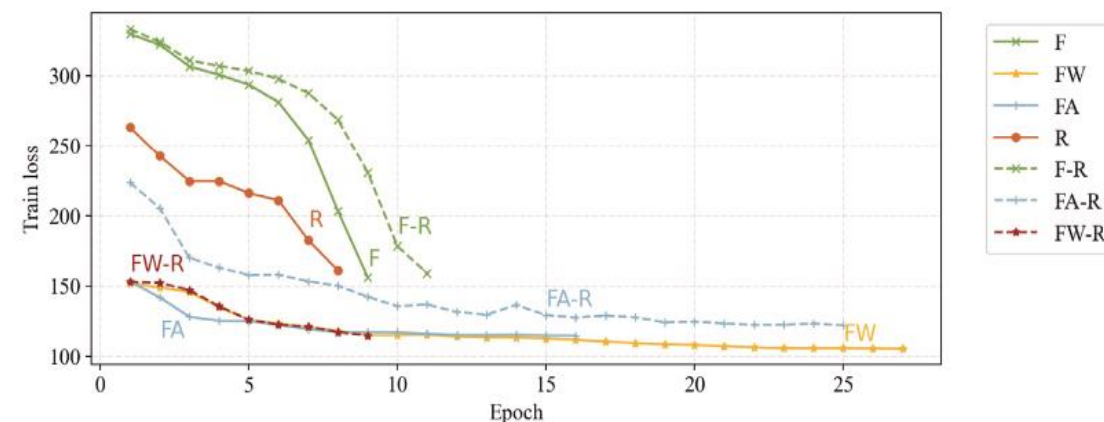


Fig. 5. Training process of different forward algorithms.

比較結果

結果の比較：予測精度の比較

2：使用アルゴリズム間の比較

2.2：アルゴリズム間のMAPE比較

Weighted Ergodic Iteration + Broyden法 が最良

Table 3
MAPE of proposed forward algorithms. MAPEs are shown in percentage and superscript.

In-distribution test								
Demand	Capacity	F	FA	FW	R	F-R	FA-R	FW-R
Base	Default	8.4	5.6 [*]	5.7	8.0	8.7	6.2	6.0
Uncongested	Default	9.5	9.1	8.1	9.5	8.3	8.5	8.0 [*]
Congested	Default	6.1	3.2	3.2	6.2	11.0	4.5	3.1 [*]
	Mean	8.0	6.0	5.7 [*]	7.9	9.3	6.4	5.7 [*]
	Std	1.8	3.0	2.4	1.7	1.4	2.0	2.5
Out-of-distribution test								
Demand	Capacity	F	FA	FW	R	F-R	FA-R	FW-R
Base	Default	7.5	5.7 [*]	5.8	7.2	7.7	6.4	6.0
	-50%	4.5	3.4 [*]	3.4 [*]	5.0	4.8	3.6	3.4 [*]
	+50%	9.1	6.9 [*]	7.0	10.2	9.3	8.8	7.4
Uncongested	Default	8.3	8.5	7.6	8.1	8.0	8.2	7.5 [*]
	-50%	9.5	8.0	7.3	9.9	7.6	14.4	6.9 [*]
	+50%	8.4	9.4	7.9 [*]	8.4	8.2	7.9 [*]	7.9 [*]
Congested	Default	5.7	3.6 [*]	3.8	5.1	10.2	4.3	3.6 [*]
	+50%	8.8	5.5 [*]	5.7	6.2	11.9	6.1	5.5 [*]
	Mean	7.7	6.4	6.1	7.5	8.5	7.5	6.0 [*]
	Std	1.8	2.2	1.7	2.0	2.1	3.4	1.7

*Denotes the best performance of each scenario.

2.3：逆伝播のアルゴリズムの比較

FA(Fixed point Approximation)が最良

JF(Jacobian-free Backpropagation)は学習に悪影響

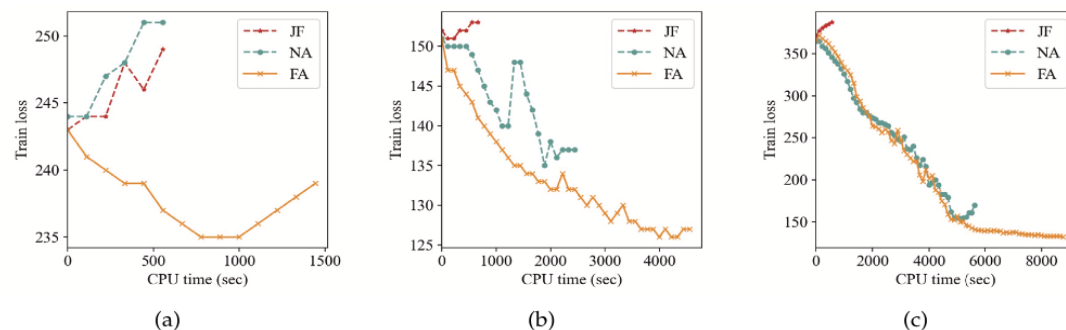


Fig. 6. Performances of different backpropagation methods under (a) base, (b) uncongested, and (c) congested demand.

2.4：スペクトル正規化による計算速度上昇評価

w : 正規化あり w/o : 正規化なし 正規化で約3~4倍早くなる

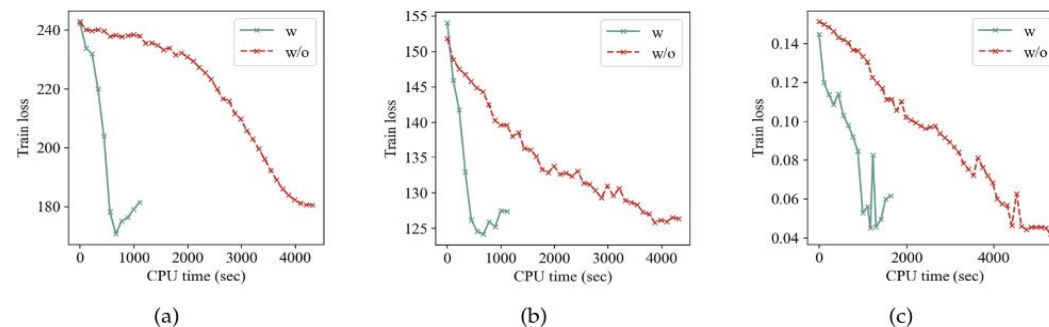


Fig. 7. Effects of spectral normalization under (a) base, (b) uncongested, and (c) congested demand. "w" suggests "with spectral normalization" and "w/o" suggests "without spectral normalization".

比較結果

結果の比較：ロバスト性の評価

3：計算のロバスト性

3.1：センサー設置の欠落

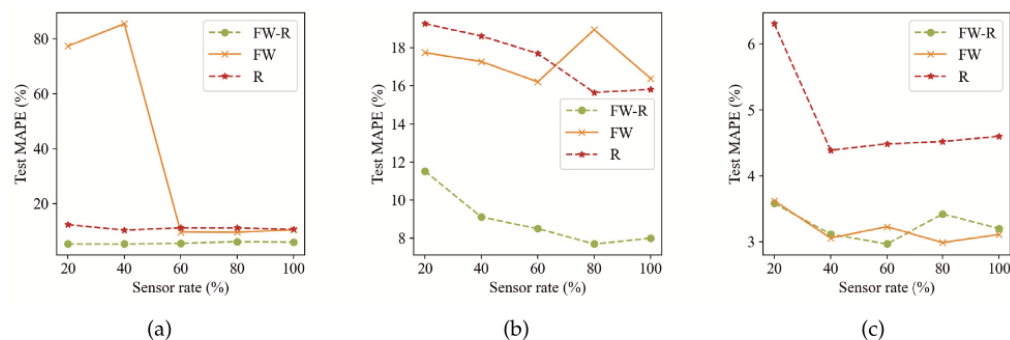


Fig. 8. Model performances with different sensor coverage rates under (a) base, (b) uncongested, and (c) congested demand.

3.2：OD需要のノイズ(誤推定)

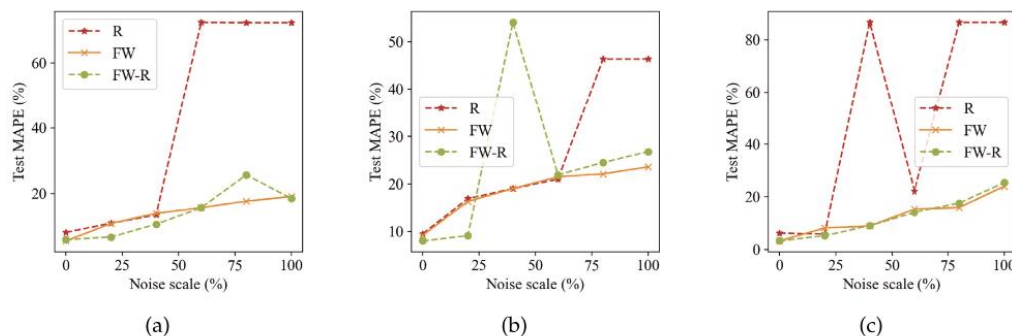


Fig. 9. Model performances with demand noises under (a) base, (b) uncongested, and (c) congested demand.

3.3：経路集合の不完全性

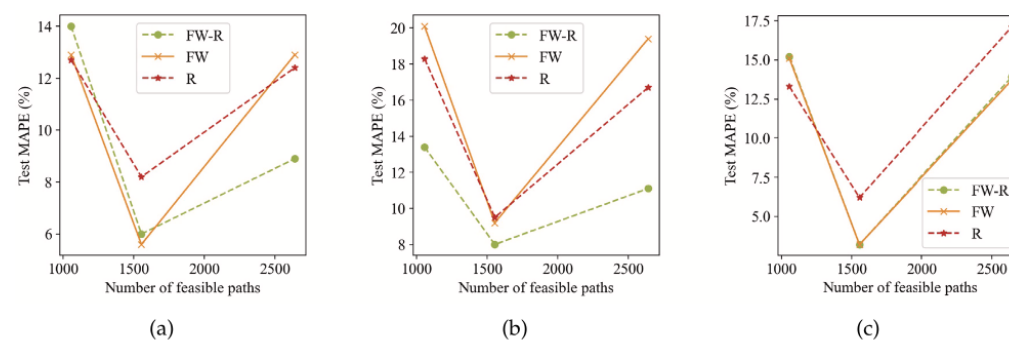


Fig. 10. Effects of inaccurate feasible path sets under (a) base, (b) uncongested, and (c) congested demand.

不完全集合 : 1058本
 選択された経路集合 : 1587本
 冗長集合 : 2645本

6 : Conclusion

結論

【有用性】

1. 観測データから直接ルートを選択選好と均衡状態を学習

- ・ネットワーク情報と個人の持つ情報を同時に考慮してNNで学習できる

2. ネットワークの変化に強い

- ・リンクが増えるごとに均衡解は変化するが、そこも次元圧縮的な概念でNNモデル内に組み込まれている

3. 不確実性へのロバスト性

- ・センサの配置率や経路集合のばらつき, 需要予測のノイズなど現実世界の問題を変数にケーススタディ

【研究の強み】

1. 均衡解の存在性を理論的に保証

- ・提案NN構造が連続性・単調性・Lipschitz性を満たす設計で均衡解の存在が定理で保証

2. 解法の安定性と収束性

- ・オペレーター分割法 + Broyden法 + 正則化でバッチ単位で安定してVIを解く
- ・収束性も理論的に議論（Appendixに詳細あり）

3. 数値実験による検証

- ・Sioux Fallsネットワークを用いたシミュレーションで既往手法(DUE, SUE)と比較し一貫して高精度

今後の課題・所感

1. 需要推定との同時学習

現在はOD需要は既知と仮定 → 将来的には「ルート選択嗜好」と「OD需要」の**同時推定**を目指す

2. 経路候補生成の最適化

初期に冗長な経路集合を与えて、**学習過程で不要な経路を削除する仕組みの実装**が課題

3. ニューラルネットのさらなる解釈性の確保

学習されたルート選択モデルの意味づけ(どの属性に重みを置いたか)を可視化・解釈可能に

1. 「ボトムアップ」アプローチからの脱出

- ・ 行動モデルそのものをデータから学習「end to end」の枠組みを提案
- ・ **行動モデルの選択(これは怪しい)**と均衡の導出を一体化

- ・ 結局行動モデルというよりは、ただの重みパラメータ最適化な感じだった.
- ・ 個々人ではなく、同一OD間の値を最頻値や平均で扱うのも違和感
- ・ 経路集合から下の均衡や、数理最適化に関してはとても深かったのに対して、行動モデルの部分が浅いのでまだ一貫したモデルとは言えない気がする

- ・ NN内に暗黙でVI制約条件を入れたのはとても面白い 他にも色々使えそう(NN型DCMの効用関数や各変数間に対する制約として)
- ・ 交通流均衡の話とこれまでの発展に関しては、かなり体系化されたと思う
- ・ 詳細な証明までは追いきれていないので頑張ろうと思います

7 : Appendix

研究の背景

ユーザー均衡の重要性と伝統的手法の限界

1 : ワードロップの第一法則の数理的表現

$$\langle C(h^*), h - h^* \rangle \geq 0 \quad \forall h \in H$$

現在の均衡配分 h^* を別の配分 h に変えた際の各経路における“コストの合計が常に正

$$\sum_{p \in P} C_p(h^*) \cdot (h_p - h_p^*) \geq 0$$

ならば

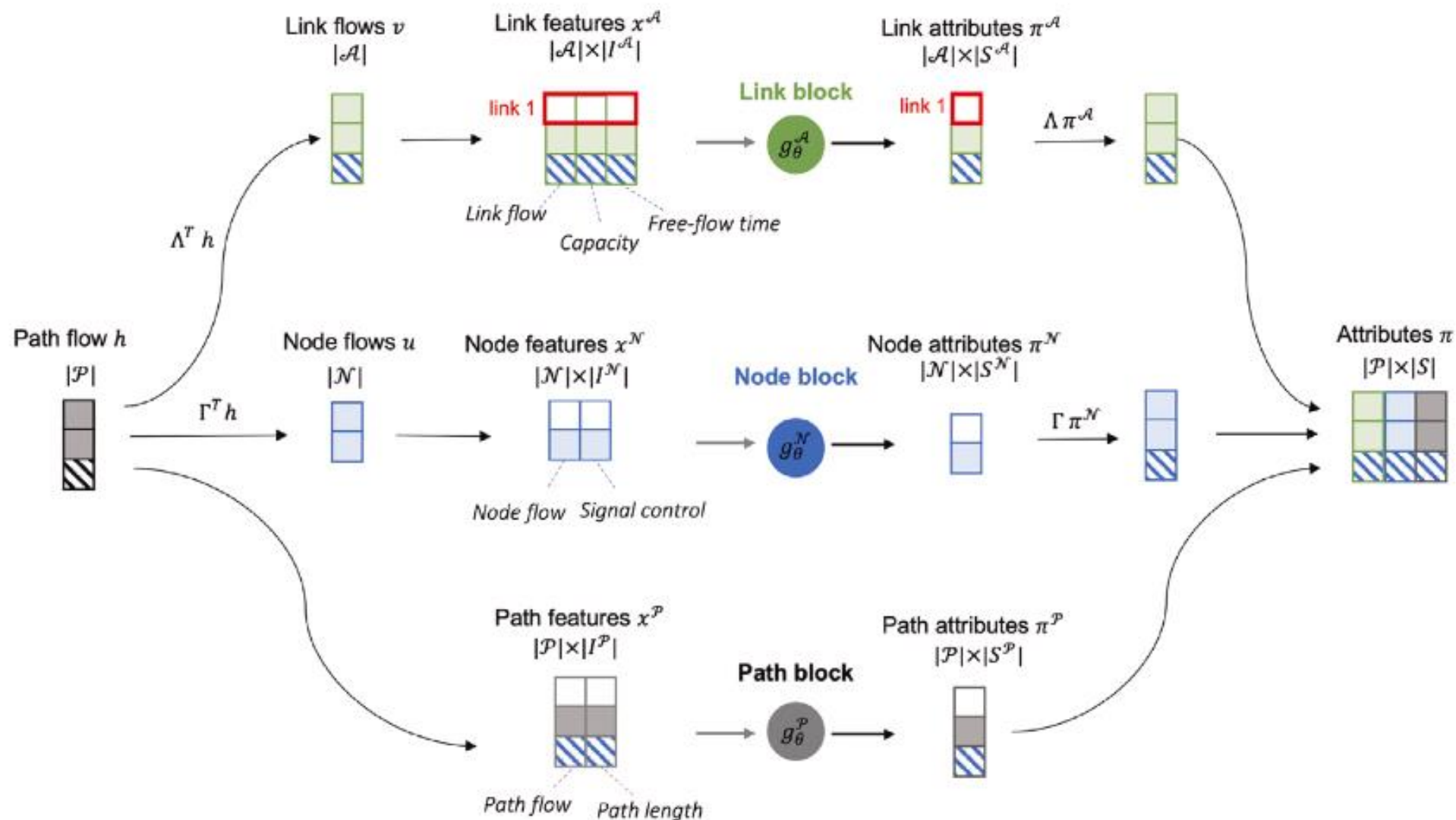
使用されていない経路のコストは同等 or 高い

$$h_p^* > 0 \rightarrow C_p(h^*) \leq C_{p'}(h^*) \quad \forall p', p' \in P_{od}$$

$$h_p^* = 0 \rightarrow C_p(h^*) \geq \min_{p' \in P_{od}, h_{p'}^* > 0} C_{p'}(h^*)$$

も成り立つ

経路集合	: P_{od}
経路	: $p', p' \in P_{od}$
経路フロー	: $h \in R^{ P }$ (非負, 需要制約を満たす)
経路コスト関数	: $C(h) \in R^{ P }$ (フローに依存)
h^*	: 均衡時の経路フロー
h	: 均衡でない時の経路フロー



固定点反復法のアルゴリズム 1

1.

初期化：

- $z^{(1)} = \tilde{z}, z^{(0)} = \tilde{z}$
- $\phi^{(0)} = \epsilon_1 + 1, k = 1$

2. while $\phi^{(k)} > \epsilon_1$ かつ $k < \kappa_1$:

- $z^{(k+1/2)} = P_{\mathcal{H}_1}(z^{(k)})$
(非負制約への射影)
- $y^{(k+1)} = P_{\mathcal{H}_2}(2z^{(k+1/2)} - z^{(k)} - \alpha_1^{(k)} C_\theta(z^{(k+1/2)}, x))$
(需要制約への射影)
- $z^{(k+1)} = z^{(k)} - z^{(k+1/2)} + y^{(k+1)}$
(補助変数の更新)
- $\phi^{(k+1)} = \|z^{(k+1)} - z^{(k)}\| / \|z^{(k)}\|$
(相対残差)
- if $\phi^{(k+1)} / \phi^{(k)} \geq \beta_1$:
 - $\alpha_1^{(k+1)} = \gamma_1 \cdot \alpha_1^{(k)}$
(ステップサイズ縮小)

3. 最終出力： $h^* = P_{\mathcal{H}_1}(z^{(k)})$

要素	内容
$T_\theta(z, x)$	オペレーター分割法によって定義された補助写像
$P_{\mathcal{H}_1}, P_{\mathcal{H}_2}$	射影演算（閉形式で定義）
射影順序	まず非負制約→その後需要制約
相対残差 $\phi^{(k)}$	収束判定指標
可行性	初期点は任意でよく、反復中に可行に近づく
目的	VIの解に対応する均衡パスフロー h^* を求める

固定点反復法のアルゴリズム2・3(高速ver)

Anderson Mixing（アンダーソン混合法）

Anderson mixingは、 τ 個の直近の反復結果の最適線形結合として $z^{(k+1)}$ を更新する：

$$z^{(k+1)} = \sum_{i=1}^{\tau} \alpha^{(i)} T_{\theta}(z^{(k-i+1)}, x)$$

ここで、係数 $\alpha^{(i)}$ は以下の**2乗最小化問題（2次計画問題）**を解くことで決定される：

$$\min_{\sum \alpha^{(i)}=1} \sum_{i=1}^{\tau} (\phi^{(k-i+1)} \alpha^{(i)})^2$$

目的は、直近 τ ステップにおける残差ノルムの総和を最小化することにある。

Broyden法

► 方程式の意味：

解きたい方程式：

$$z = T_{\theta}(z, x) \iff z - T_{\theta}(z, x) = 0$$

これは 非線形方程式のゼロ点（root）を探す問題。

► Newton法との違い：

通常の Newton法：

$$z^{(k+1)} = z^{(k)} - J^{-1}(z^{(k)}) \cdot (z^{(k)} - T_{\theta}(z^{(k)}))$$

→ ヤコビ行列 J を明示的に使う必要がある

→ 本研究の射影 $P_{\mathcal{H}_1}$ は微分不可能なため、適さない

Weighted Ergodic Iteration（重み付きエルゴード反復）

Weighted ergodic iteration は、毎ステップでステップサイズをヒューリスティックに調整しながら、 $z^{(k)}$ を次のように更新する：

$$z^{(k)} = \frac{2}{(k+1)(k+2)} \sum_{i=0}^k (i+1) z^{(i)}$$

この手法により、固定点反復の収束レートを

$$O\left(\frac{1}{\sqrt{k+1}}\right) \rightarrow O\left(\frac{1}{k+1}\right)$$

に改善することができる（ただしコスト関数のヤコビアンが対称であるとき）。これはAppendix Bで証明されており、Attribute Net が式 (6) に従って構築されていればこの条件は成り立つ。

サマリー

- 固定点反復：最も基本、まず使う手法
- Anderson Mixing：初期収束が遅いときの加速用
- Weighted Ergodic：理論保証を重視するときの安定化手法
- Broyden法：高精度 or 収束困難なケースに対応した強力手法