Uncertain Constraints in Imitation
and Reinforcement Learning

Matthew Howard

Institute for Perception, Action and Behaviour,
University of Edinburgh

12 November 2009
Motion constraints are **ubiquitous** in everyday behaviour.

**Examples:** opening a door, wiping a surface, carrying an object, maintaining balance. . .

If we want robots to learn such tasks effectively, they must be able to **handle constraints appropriately**.
Motion constraints are **ubiquitous** in everyday behaviour.

**Examples:** opening a door, wiping a surface, carrying an object, maintaining balance...

If we want robots to learn such tasks effectively, they must be able to **handle constraints appropriately**.

**Uncertainty in Constraints**

- Exact modelling of the environment often **infeasible**.
- Task constraints are often **ambiguous** or **hard to formalise**.
Motion constraints are **ubiquitous** in everyday behaviour.

**Examples**: opening a door, wiping a surface, carrying an object, maintaining balance...

If we want robots to learn such tasks effectively, they must be able to **handle constraints appropriately**.

---

**Uncertainty in Constraints**

- Exact modelling of the environment often **infeasible**.
- Task constraints are often **ambiguous** or **hard to formalise**.

How can we deal with **uncertain constraints** in our robot learning schemes?
Outline

1. Learning from Data with Unknown Constraints
   - Abstraction of Motion in Imitation Learning
   - Methods for Constraint-consistent Learning

2. Learning the Constraint
   - Extracting and Exploiting Constraint Information
   - Learnt Constraints in Reinforcement Learning
Outline

1 Learning from Data with Unknown Constraints
   - Abstraction of Motion in Imitation Learning
   - Methods for Constraint-consistent Learning

2 Learning the Constraint
   - Extracting and Exploiting Constraint Information
   - Learnt Constraints in Reinforcement Learning
Abstraction of Motion

Many everyday behaviours can be framed in terms of performing some **task** in the presence of **context-specific constraints**.

- Window cleaning involves wiping.
- Maintaining balance on a ladder constrains possible movements.

Given examples of movements, can we find a representation that captures the **essence of the task** but ignores the specific constraints?

[Takano et al., 2006, Grasse, 2005, Inamura et al., 2004]
Learning Policies from Constrained Movements

Basic problem: Given data \((x_n, u_n)\), learn the policy \(\pi : x \rightarrow u\).

e.g. [Calinon and Billard, 2007, Grimes et al., 2007, Schaal et al., 2003]
Learning Policies from Constrained Movements

Basic problem: Given data \((x_n, u_n)\), learn the policy \(\pi: x \rightarrow u\).

e.g. [Calinon and Billard, 2007, Grimes et al., 2007, Schaal et al., 2003]

Policy: ‘extend finger’ without and with obstacle constraint.

Constraints:

- Modify appearance and distribution of observations.
- Usually not explicitly observable.
- May change dynamically between or during observations.

Problematic for standard learning approaches (model averaging, multiple models required).
Given example movements performed under a variety of constraints.

Learn the unconstrained policy.

Generalise predict behaviour under new, unseen constraints.
**Constrained Motion Model**

**Constraint Model** [Udwadia, 2008, Peters et al., 2008]

- Policy $\pi(x)$, subject to constraints $A(x, t)\pi = 0$
- $A(x, t)$ unknown, and varies between/during observations.
- Observe constrained actions:

  $$u = N(x, t)\pi(x) \quad \text{where} \quad N(x, t) = I - A^\dagger A$$

**Hard constraints**, for example:

- interaction with environment: e.g. opening a door.
- self-imposed constraints (‘task constraints’).
Constraints eliminate components of the action in certain (constrained) directions of the space.

Best generalisation → can constrain in **maximum** number of ways with **minimum** error.

**Error measures:**

- **Constrained policy error (CPE)**
  \[ E_{cpe}[\tilde{\pi}] = \sum_{n=1}^{N} \| u_n - N_{\tilde{\pi}}(x_n) \|_2 \]

- **Unconstrained policy error (UPE)**
  \[ E_{upe}[\tilde{\pi}] = \sum_{n=1}^{N} \| \pi_n - \tilde{\pi}(x_n) \|_2 \]
Measure of Success

Constraints eliminate components of the action in certain (constrained) directions of the space.

Best generalisation → can constrain in **maximum** number of ways with **minimum** error.

Error measures:

1. **Constrained policy error (CPE)**
   
   \[ E_{cpe}[\tilde{\pi}] = \sum_{n=1}^{N} \| u_n - N_n \tilde{\pi}(x_n) \|^2 \]

2. **Unconstrained policy error (UPE)**
   
   \[ E_{upe}[\tilde{\pi}] = \sum_{n=1}^{N} \| \pi_n - \tilde{\pi}(x_n) \|^2 \]
Direct Learning

Minimise:

\[ E_{direct}[\tilde{\pi}] = \sum_{n=1}^{N} \| u_n - \tilde{\pi}(x_n) \|^2 \]

- Simplest approach → regress directly on observed \(x, u\).
- Effective for **unconstrained/consistently constrained** \(\pi\).

[Calinon and Billard, 2007, Hersch et al., 2006, Schaal et al., 2003, Ijspeert et al., 2003]
Direct Learning

Minimise:

\[ E_{direct}[\tilde{\pi}] = \sum_{n=1}^{N} \| u_n - \tilde{\pi}(x_n) \|^2 \]

- **Naive** to the effect of variable constraints.

- Simplest approach → regress directly on observed \( x, u \).

- Effective for \textbf{unconstrained/consistently constrained} \( \pi \).

[Calinon and Billard, 2007, Hersch et al., 2006, Schaal et al., 2003, Ijspeert et al., 2003]
Direct Learning

Minimise:

\[ E_{\text{direct}}[\tilde{\pi}] = \sum_{n=1}^{N} \| u_n - \tilde{\pi}(x_n) \|^2 \]

- Naive to the effect of variable constraints.
  → Model averaging

- Simplest approach → regress directly on observed \( x, u \).
- Effective for **unconstrained/consistently constrained** \( \pi \).

[Calinon and Billard, 2007, Hersch et al., 2006, Schaal et al., 2003, Ijspeert et al., 2003]
Measure of Success: Revisited

1. Unconstrained policy error (UPE)
   \[ E_{upe}[\tilde{\pi}] = \sum_{n=1}^{N} \| \pi_n - \tilde{\pi}(x_n) \|^2 \]

2. Constrained policy error (CPE)
   \[ E_{cpe}[\tilde{\pi}] = \sum_{n=1}^{N} \| u_n - N_n \tilde{\pi}(x_n) \|^2 \]
Measure of Success: Revisited

1. Unconstrained policy error (UPE)
   \[ E_{upe}[\tilde{\pi}] = \sum_{n=1}^{N} \| \pi_n - \tilde{\pi}(x_n) \|^2 \]
   \[ \rightarrow \text{unknown} \]

2. Constrained policy error (CPE)
   \[ E_{cpe}[\tilde{\pi}] = \sum_{n=1}^{N} \| u_n - N_n \tilde{\pi}(x_n) \|^2 \]
   \[ \rightarrow \text{unknown} \]
Measure of Success: Revisited

1. Unconstrained policy error (UPE)
   \[ E_{upe}[	ilde{\pi}] = \sum_{n=1}^{N} \| \pi_n - \tilde{\pi}(x_n) \|^2 \]
   \[ \rightarrow \text{unknown} \]

2. Constrained policy error (CPE)
   \[ E_{cpe}[	ilde{\pi}] = \sum_{n=1}^{N} \| u_n - N_n \tilde{\pi}(x_n) \|^2 \]
   \[ \rightarrow \text{unknown} \]

next best thing. . .

3. ‘Inconsistency error’
   \[ E_i[	ilde{\pi}] = \sum_{n=1}^{N} \| u_n - \hat{u}_n \hat{u}_n^T \tilde{\pi}(x_n) \|^2 \]

Motivation
Given multiple observed actions \( u \) at point \( x \), the projection of the policy onto any \( u \) must \textit{match} that \( u \),

\[ u \dashv \hat{u}^T \pi(x) \]
Measure of Success: Revisited

1. Unconstrained policy error (UPE)
   \[ E_{upe}[\tilde{\pi}] = \sum_{n=1}^{N} \| \pi_n - \tilde{\pi}(x_n) \|^2 \]
   \[ \rightarrow \text{unknown} \]

2. Constrained policy error (CPE)
   \[ E_{cpe}[\tilde{\pi}] = \sum_{n=1}^{N} \| u_n - N_n \tilde{\pi}(x_n) \|^2 \]
   \[ \rightarrow \text{unknown} \]

   \[ \downarrow \text{next best thing...} \]

3. ‘Inconsistency error’
   \[ E_i[\tilde{\pi}] = \sum_{n=1}^{N} \| u_n - \hat{u}_n \hat{u}_n^T \tilde{\pi}(x_n) \|^2 \]
   \[ \rightarrow \text{only requires observed } x, u. \]

Motivation

Given multiple observed actions \( u \) at point \( x \), the projection of the policy onto any \( u \) must \textit{match} that \( u \),

\[ u \parallel \hat{u} \hat{u}^T \pi(x) \]
Algorithms: Inconsistency Approach

**Inconsistency Approach** [Howard et al., 2009a]

Project onto observations, optimise

\[
E_i[\tilde{\pi}] = \sum_{n=1}^{N} \| u_n - \hat{u}_n \hat{\pi}_n^T \tilde{\pi}(x_n) \|^2
\]

- Penalises models that are **inconsistent** with the observations.
- Can incorporate into well-known policy models, e.g.

**Basis Function Models**

Policy model: \( \tilde{\pi}(x) = Wb(x) \)

- e.g. Linear model \( b(x) = (x^T, 1)^T \),
- Normalised RBFs

\[
b_i(x) = \frac{K(x-c_i)}{\sum_{j=1}^{M} K(x-c_j)}
\]

Can directly solve for optimal weights \( \rightarrow \) efficient batch training.

**Locally Linear Models**

Policy model: \( \tilde{\pi}(x) = \frac{1}{W} \sum_{m=1}^{M} w_m \tilde{\pi}_m(x) \)

- Local models \( \tilde{\pi}_m(x) = B_m(x^T, 1)^T \)
- Receptive fields

\[w_m = \exp \left( -\frac{1}{2\sigma^2} \| x - c_m \|^2 \right)\]

Local optimisation of \( E_i[\tilde{\pi}] \)
\( \rightarrow \) better for high dimensional \( x, u \).
**Other Methods**

### Alignment Approach [Howard et al., 2008]

Integrate, align, learn potential.

- Raw data: constrained trajectories.
- Estimate $X_k, \hat{\Phi}_k$ using Euler integration.
- Find optimal offsets $b_{opt} = -H^\dagger a$ [Verbeek, 2006]
- Train global model on data tuples $(x_{kn}, \hat{\phi}_{kn} + b_{opt}^k)$

- Quadratic ‘min.-disagreement’ objective function $\rightarrow$ efficient optimisation.
- Can use eigenvalues of $H$ to determine ‘outlier’ trajectories.
Hybrid Approach [Howard et al., 2009b]

Two-step optimisation:

1. **Primary optimisation of inconsistency**
   \[ E_1[\tilde{\pi}] = \sum_{n=1}^{N} \| \mathbf{u}_n - \hat{\mathbf{u}}_n \hat{\mathbf{u}}_n^T \tilde{\pi}(\mathbf{x}_n) \|^2 \]

2. **Secondary optimisation of standard risk**
   \[ E_2[\tilde{\pi}] = \sum_{n=1}^{N} \| \mathbf{u}_n - \tilde{\pi}(\mathbf{x}_n) \|^2 \]
   s.t. \[ \tilde{\pi} \in \arg \min_{\pi'} \{ E_i[\pi'] \} \]
Hybrid Approach [Howard et al., 2009b]

Two-step optimisation:

1. Primary optimisation of inconsistency

\[ E_i[\tilde{\pi}] = \sum_{n=1}^{N} \| u_n - \hat{u}_n \hat{\pi}^T(x_n) \|^2 \]

2. Secondary optimisation of standard risk

\[ E_2[\tilde{\pi}] = \sum_{n=1}^{N} \| u_n - \tilde{\pi}(x_n) \|^2 \]

s.t. \( \tilde{\pi} \in \text{arg min}_{\pi'} \{ E_i[\pi'] \} \).

Key Idea

- Use eigenvalues of \( H \) to find significant elements.
- Parametric and local linear learners derived.
- Closed form solution \( \rightarrow \) efficient optimisation.

Optimise \( E_2[\tilde{\pi}] \) in nullspace of inconsistency optimisation.
Hybrid Approach [Howard et al., 2009b]

Two-step optimisation:

1. Primary optimisation of inconsistency
   \[ E_1[\tilde{\pi}] = \sum_{n=1}^{N} \| u_n - \hat{u}_n \hat{u}_n^T \tilde{\pi}(x_n) \|^2 \]
   \[ \rightarrow \text{constraint-consistent learning.} \]

2. Secondary optimisation of standard risk
   \[ E_2[\tilde{\pi}] = \sum_{n=1}^{N} \| u_n - \tilde{\pi}(x_n) \|^2 \]
   s.t. \[ \tilde{\pi} \in \arg\min_{\pi'} \{ E_i[\pi'] \} \]
   \[ \rightarrow \text{use free parameters to tighten fit.} \]

- Use eigenvalues of \( H \) to find significant elements.
- Parametric and local linear learners derived.
- Closed form solution \( \rightarrow \) efficient optimisation.

Key Idea

Optimise \( E_2[\tilde{\pi}] \) in nullspace of inconsistency optimisation.
Experimental Goals

**Toy Example**
- Validate the theory, illustrate the concepts.
- Test performance in terms of
  - policy complexity, data requirements, noise robustness.

**Grasping Example**
- Test **scalability**
  - for realistic policy and constraints
  - for higher dimensional $x, u$
- Investigate **generalisation** over constraints

**Learning to Wash a Car from Human Data**
Test the approach for representing/reproducing human movements.
Experimental Goals

**Toy Example**
- Validate the theory, illustrate the concepts.
- Test performance in terms of
  - policy complexity, data requirements, noise robustness.

**Grasping Example**
- Test **scalability**
  - for **realistic policy and constraints**
  - for **higher dimensional** $x, u$
- Investigate **generalisation** over constraints

**Learning to Wash a Car from Human Data**
Test the approach for representing/reproducing human movements.
Demo Scenario: Grasping a ball in a cluttered environment.

‘Ball grasping’ policy:

- Inverted Gaussian potential
  \[ \phi(x) = \alpha \left( 1 - e^{\|x - x_c\|^2 / 2\sigma^2} \right). \]

Constraints:

- Barriers, random gap widths.

- Sampling: 50 Hz → 100 trajectories, 100 points per trajectory.

- Policy model: 150 Gaussian RBFs, centres placed by k-means.
Unconstrained reaching: Expert, direct & inconsistency policies.

<table>
<thead>
<tr>
<th>Constraint</th>
<th>Direct</th>
<th>Incon.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Training</td>
<td>0.0531 ± 0.0068</td>
<td>0.0052 ± 0.0022</td>
</tr>
<tr>
<td>Unseen Barrier</td>
<td>0.4630 ± 0.0350</td>
<td>0.0052 ± 0.0022</td>
</tr>
<tr>
<td>Unconstrained</td>
<td>0.9216 ± 0.0625</td>
<td>0.0052 ± 0.0022</td>
</tr>
</tbody>
</table>

- **Direct** approach reproduces the constraint unnecessarily.
- Suboptimal path, fails to get ball under new constraints e.g. barrier in front of ball.
Learning to Wash a Car

Car Washing Task

Simple ‘wiping’ policy subject to varying constraints:

- **physical constraint** (hand cannot penetrate surface)
- **self-imposed constraint** (no lifting of sponge)

Different shapes and orientations of surfaces → complex, non-linear constraints.
Learning to Wash a Car

Experimental Setup

- Human wiping demonstrations on a perspex sheet.
- 3-D sponge position tracked with stereo vision (20fps)
- Sheet oriented flat, tilted $\pm 16^\circ$, $\pm 27^\circ$ about $x$-axis and $\pm 16^\circ$ about $y$-axis.

- Policy model: 300 RBFs with centres placed by $k$-means.
- 3-D mapping from hand positions to velocities.
**Motivation**

Coping with variable motion constraints in learning:
- **Naive** methods not adequate for this class of problems.

**Constraint-consistent Learning**

Recovers (unconstrained) policy from constrained observations:
- Allows us to generalise to **unseen constraints**.
- Extracts ‘essence’ of movement from specific examples.

**Extensions**

- Extension to alternative constraint types.
- Learning force/torque control policies, dynamics models.
1. Learning from Data with Unknown Constraints
   - Abstraction of Motion in Imitation Learning
   - Methods for Constraint-consistent Learning

2. Learning the Constraint
   - Extracting and Exploiting Constraint Information
   - Learnt Constraints in Reinforcement Learning

Uncertain Constraints in Imitation, and Reinforcement Learning
Benefits of Learning the Constraint

Automatically decompose movements into policy and constraint

- more accurate, flexible models of observed behaviour.
Benefits of Learning the Constraint

Automatically decompose movements into policy and constraint

- more accurate, flexible models of observed behaviour.

So far, assumed the constraints are unknown for learning.

- Given $N$ could use the CPE to improve model accuracy.
Benefits of Learning the Constraint

Automatically decompose movements into policy and constraint

- more accurate, flexible models of observed behaviour.

So far, assumed the constraints are unknown for learning.

- Given N could use the CPE to improve model accuracy.

Additionally, knowing the constraint, can further optimise the movement with respect to the task.
**Benefits of Learning the Constraint**

**Example: Carrying a ball to a target.**

- Task constraint: Successful carrying requires that the hands remain at fixed separation.
Benefits of Learning the Constraint

Example: Carrying a ball to a target.

- Task constraint: Successful carrying requires that the hands remain at fixed separation.
Benefits of Learning the Constraint

Example: Carrying a ball to a target.

- Task constraint: Successful carrying requires that the hands remain at fixed separation.
- If we can enforce the constraint up-front, learning the task is simplified.
Benefits of Learning the Constraint

Example: Carrying a ball to a target.

- Task constraint: Successful carrying requires that the hands remain at fixed separation.
- If we can enforce the constraint up-front, learning the task is simplified.

How can we extract this information from our demonstrations?
Analytically, can identify a two-dimensional holonomic constraint:

\[ r_1 = r_3 + \delta; \quad r_2 = r_4 \]

Four-dimensional system only has two remaining DOFs.

\[ \rightarrow \text{Successful movements must lie on this manifold.} \]
Learning the Constraint with DR

Analytically, can identify a two-dimensional holonomic constraint:

\[ r_1 = r_3 + \delta; \quad r_2 = r_4 \]

Four-dimensional system only has two remaining DOFs.

→ Successful movements must lie on this manifold.

**Key Idea** [Bitzer et al., 2009]

- Use dimensionality reduction to find the two-dimensional space in which the constraint is satisfied.
Learning the Constraint with DR

Analytically, can identify a two-dimensional holonomic constraint:

\[ r_1 = r_3 + \delta; \quad r_2 = r_4 \]

Four-dimensional system only has two remaining DOFs.

→ Successful movements must lie on this manifold.

**Key Idea** [Bitzer et al., 2009]

- Use dimensionality reduction to find the two-dimensional space in which the constraint is satisfied.

→ Focus learning on solving other aspects of the task without worrying about the constraint.
- Reduces the size of the planning problem.
- **Avoids expensive exploration** away from the constraint manifold.
Reinforcement Learning in Latent space

Combining RL with DR

Simply replace the **full state space** with the **learnt latent space**.

- Each step in LS $\rightarrow$ new target reference in FS.
Experiments

**Dimensionality Reduction**

Gaussian Process Latent Variable Model (GPLVM) [Lawrence, 2005]
- Non-linear, probablistic DR technique.
- Good generalisation on sparse data.

**Reinforcement Learning**

RL using TD(0) V-function learning [Sutton and Barto, 1998, Neumann, 2005]
- Robust, online learning without the need for careful initialisation/tuning of parameters.

**Test Cases**
- Bi-manual ball-carrying in end-effector- and joint-space.
- Obstacle avoidance on the KUKA 7-DOF arm.
Experiments

Dimensionality Reduction

Gaussian Process Latent Variable Model (GPLVM) [Lawrence, 2005]
- Non-linear, probabilistic DR technique.
- Good generalisation on sparse data.

Reinforcement Learning

RL using TD(0) V-function learning [Sutton and Barto, 1998, Neumann, 2005]
- Robust, online learning without the need for careful initialisation/tuning of parameters.

Test Cases

- Bi-manual ball-carrying in **end-effector**- and **joint-space**.
- Obstacle avoidance on the KUKA 7-DOF arm.
Bi-manual Reaching in End-effector space

Setup

- **Reward**: Gaussian centered on target.
- **Actions**: $\pm 0.1 \text{ m}$ in each dimension of latent/full space.
- $V^\pi$ represented as $20^n$ grid of RBFs, random initial weights.
- Episodes terminate if (i) constraint is broken, (ii) obstacle is hit, (iii) state-space boundaries are hit.

Compare Learning

- in end-effector space $x \equiv r \in \mathbb{R}^4$
- in latent space $z \in \mathbb{R}^2$ of GPLVM (trained on 200 data pts satisfying the constraints, $\delta = 0.1 \text{ m}$)
Bi-manual Reaching in Task-space

- RL in the full space naive to the constraint, must learn to satisfy by exploration.
- DR-RL explores more of the relevant space → much faster convergence to a solution.
Bi-manual Reaching in Joint-space

Setup

- Full-space: \( x \equiv q \in \mathbb{R}^4 \) joint angles
- GPLVM on 1000 data pts, 2D latent space.
- Reward defined as Gaussian around target in task space.
**Setup**

- **Full-space:** \( \mathbf{x} \equiv \mathbf{q} \in \mathbb{R}^4 \) joint angles
- **GPLVM** on 1000 data pts, 2D latent space.
- **Reward** defined as Gaussian around target in task space.

<table>
<thead>
<tr>
<th></th>
<th>Bimanual TS</th>
<th>Bimanual JS</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>PCA</strong></td>
<td>0.05 ± 0.01</td>
<td>3.25 ± 0.41</td>
</tr>
<tr>
<td><strong>GPLVM</strong></td>
<td>0.24 ± 0.18</td>
<td>0.80 ± 0.70</td>
</tr>
<tr>
<td><strong>PCA</strong></td>
<td>100.00 ± 0.00</td>
<td>4.92 ± 0.81</td>
</tr>
<tr>
<td><strong>GPLVM</strong></td>
<td>94.50 ± 5.61</td>
<td>61.03 ± 6.16</td>
</tr>
</tbody>
</table>
Bi-manual Reaching in Joint-space

**Setup**

- **Full-space:** \( x \equiv q \in \mathbb{R}^4 \) joint angles
- **GPLVM on 1000 data pts, 2D latent space.**
- **Reward defined as Gaussian around target in task space.**

![Graph and diagrams showing performance over episodes]
Learning on a 7-DOF KUKA arm

- Full-space: \( \mathbf{x} \equiv \mathbf{q} \in \mathbb{R}^7 \) joint angles
- GPLVM on 1000 data pts, 2D latent space.
- TD(0) learning, \( V^\pi \) weights randomly initialised.
- Learns to reach around the box to the target with minimal prior knowledge.
Uncertain Constraints

Coping appropriately with uncertain constraints is important in many imitation and reinforcement learning problems.

Constraint-consistent Learning

Recovers (unconstrained) policy from constrained observations:
- Allows us to generalise to unseen constraints.
- Applications in behaviour recognition and transfer.

Learning Constrained-space Models

Non-linear DR captures the manifold where constraints are satisfied in a data-driven way.
- Reduces the size of the planning problem.
- No wasteful exploration of regions where no solution exists.
Acknowledgements

Thank you

My collaborators... 

- Sebastian Bitzer
- Stefan Klanke
- Sethu Vijayakumar
- Michael Gienger
- Christian Goerick & others at HRI-EU

...and you for your attention.
Using dimensionality reduction to exploit constraints in reinforcement learning.

Learning of gestures by imitation in a humanoid robot.

Lifting of trajectories of control systems related by smooth mappings.

Learning nonparametric models for probabilistic imitation.

Learning dynamical system modulation for constrained reaching tasks.

Learning potential-based policies from constrained motion.
In *IEEE International Conference on Humanoid Robots*.

A novel method for learning policies from constrained motion.
In *IEEE International Conference on Robotics and Automation*.

Robust constraint-consistent learning.
In *IEEE International Conference on Intelligent Robots and Systems*.

Learning attractor landscapes for learning motor primitives.


Learning non-linear image manifolds by combining local linear models.

*IEEE Transactions on Pattern Analysis & Machine Intelligence, 28(8):1236–1250.*