THE UNIVERSITY OF MELBOURNE

IJCAI-17 MELBOURNE

Adversarial Generation of Real-time Feedback with Neural Networks for Simulation-based Training

Xingjun Ma, Sudanthi Wijewickrema, Shuo Zhou, Yun Zhou, Zakaria Mhammedi, Stephen O'Leary, James Bailey

xingjunm@student.unimelb.edu.au

School of Computing and Information Systems, The University of Melbourne

Introduction

- Virtual Reality (VR) provides a risk-free, low-cost and convenient training platform for many application domains.
- Real-time performance feedback is essential in such training in order to facilitate efficient task/knowledge learning.
- Here, we use the adversarial technique in neural networks to generate feedback in VR temporal bone surgery.

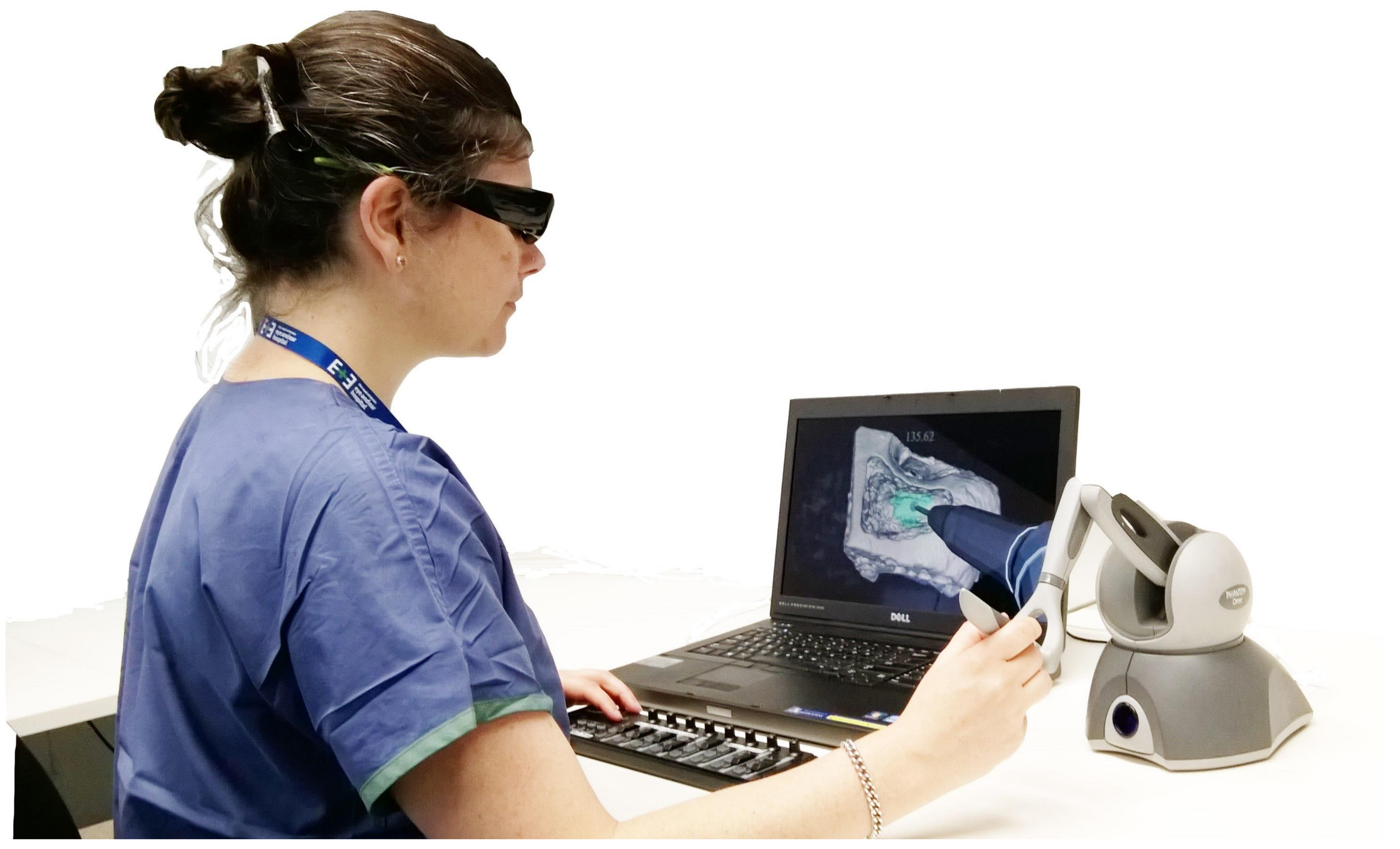


Figure 1: *The University of Melbourne Virtual Reality Temporal Bone Surgery Simulator: it consists of a computer that runs a 3D model of a human temporal (ear) bone and a haptic device that provides tactile resistance to simulate drilling.*

Feedback Generation Problem

- Given a neural network $\mathrm{N}(x)$ and a novice instance x_0, the problem is to find the optimal action $A: x_0 \rightarrow x$ that changes x_0 to an instance x under limited cost C such that x has the highest probability of being in the expert class:

$$\operatorname*{argmax}_{x} N(x), \qquad \text{subject to } loss(x_0, x) < \mathrm{C}$$

  where, feedback $A: x_0 \rightarrow x$ involves one or more feature changes (increase/decrease).

- For example, the action $A: (force = 0.2,\ speed = 0.3) \rightarrow (force = 0.5,\ speed = 0.3)$ translates to the feedback "**increase force to 0.5**".
- In VR training, the number of feature changes should be kept low to decrease cognitive load and avoid distraction:

$$loss(x_0, x) = \|x_0 - x\|_0$$

☐ **Note:** This is different to typical adversarial examples in that it targets high confidence improvements with a lesser number of perturbations.

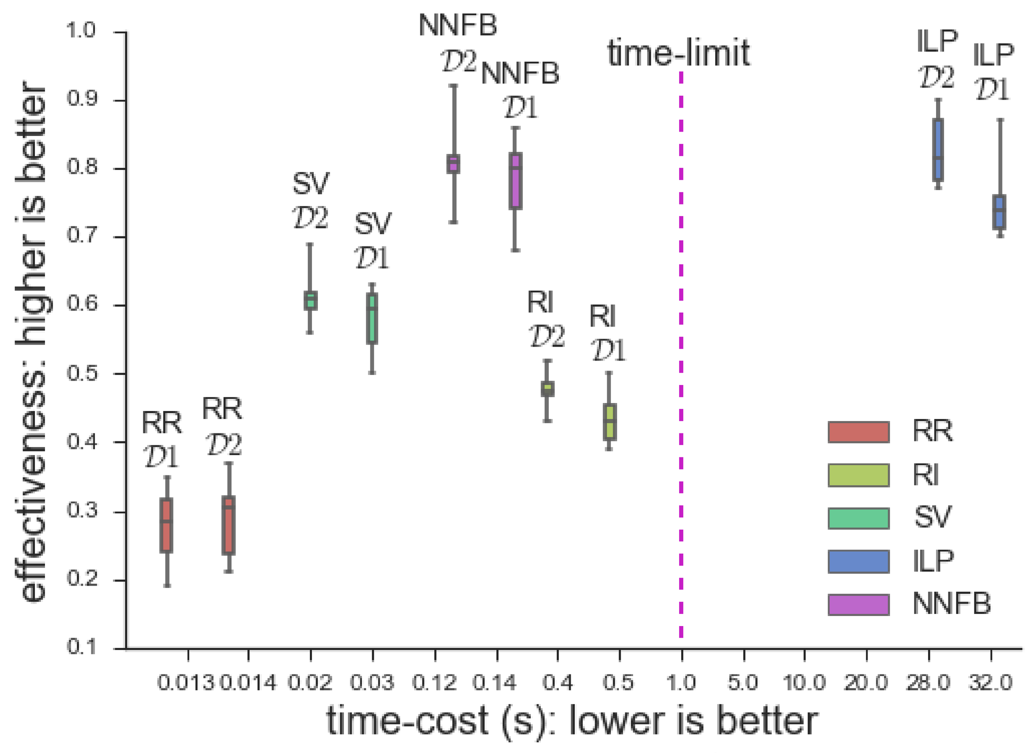
Neural Network Feedback

- **Step 1:** Pre-train a neural network classifier offline with loss $J_\Theta(x, y)$, via supervised learning.
- **Step 2:** For a real-time novice skill vector x_0 and a target (expert) skill level y^*, adversarially perturb x_0 iteratively:

$$x = x_0$$
$$x = x - \varepsilon S_x\left(x S_x - \frac{\boldsymbol{a}}{2}(1 + S_x) + \frac{\boldsymbol{b}}{2}(1 - S_x)\right)$$
$$S_x = \mathrm{sign}(\nabla_x(J_\Theta(x, y^*) + \lambda\|x - x_0\|_1))$$

- **Step 3:** clip away small changes and generate feedback: $A: x_0 \rightarrow x$
- **Step 4:** Deliver *feedback* to trainee in the form of audio instructions in order to enhance performance.

Experimental Results



✓ High confidence

✓ High effectiveness

✓ Low time-cost

✓ Suitable for real-time

Figure 2: *Comparison results (effectiveness vs time). $\mathcal{D}_1$ and $\mathcal{D}_1$ are two datasets.* ***NNFB is the proposed method.***

Table 1: *Comparison of effectiveness (mean±std). Best results are in bold.*

		NN	RF	LR	SVM	NB	KNN
$\mathcal{D}1$	RR	0.19±0.06	0.23±0.10	0.35±0.07	0.27±0.06	0.32±0.12	0.30±0.05
	RI	0.44±0.07	0.39±0.04	0.50±0.08	0.46±0.06	0.42±0.12	0.40±0.08
	SV	0.63±0.07	0.59±0.06	0.60±0.07	0.62±0.06	0.50±0.11	0.53±0.07
	ILP	0.72±0.04	**0.87±0.00**	0.71±0.05	0.76±0.04	**0.70±0.11**	**0.76±0.04**
	NNFB	**0.86±0.01**	0.82±0.08	**0.78±0.05**	**0.82±0.04**	0.68±0.14	0.73±0.08
$\mathcal{D}2$	RR	0.21±0.04	0.22±0.07	0.29±0.04	0.37±0.02	0.32±0.11	0.32±0.06
	RI	0.48±0.04	0.49±0.04	0.47±0.09	0.52±0.05	0.47±0.12	0.43±0.10
	SV	0.61±0.08	0.69±0.04	0.62±0.05	0.61±0.07	0.56±0.11	0.59±0.04
	ILP	0.88±0.04	**0.90±0.02**	0.79±0.07	**0.77±0.03**	0.78±0.12	**0.84±0.09**
	NNFB	**0.92±0.02**	0.82±0.06	**0.81±0.07**	0.72±0.05	**0.79±0.11**	0.81±0.07

Conclusions

- Adversarial perturbation can be used to generate confident expert-like feedback efficiently.
- The proposed L_0 regularization perturbation generates simple yet confident feedback.
- The proposed bounded perturbation provides flexible control over skill vectors.