Providing Automatic Feedback to Trainees after Automatic Evaluation

Mégane Millan, Catherine Achard

Providing Automatic Feedback to Trainees after Automatic Evaluation

Mégane Millan[1] and Catherine Achard[1]

Abstract— **Learning how to perform precise and controlled gestures is difficult, especially when feedback about made errors is sparse. Therefore, some works try to facilitate learning by providing virtual "*coaches*". Most of them propose to automatically score task quality. But simply assessing quality through a score is not enough. Indeed, it is essential to provide explanations on assigned scores just like experts do when supervising trainees. However when quality assessment is done automatically, such explanations are rare and computing an automatic feedback is complex. In this work, we propose to address this problem by providing an automatic feedback based on neural network explanation. Contrary to previous state of the art methods, which are focused on neural networks explicability for classification tasks, we want to explain network decision on a regression problem (quality score prediction). Thus, we propose to use gradient-based approaches and adapt them to a regression task. Moreover, to address the problem of noise present in sensitivity maps, we propose a solution that leads to more robust gradients. To test our approach, since automatic quality assessment datasets do not contain ground truth about errors position and amplitude, a synthetic dataset representing a simple temporal task has been created, with its associated ground truth. Once the method has been validated on this synthetic dataset, we apply it on real data composed of robotic surgical gestures.**

I. INTRODUCTION

Surgical training is evolving by including simulation and virtual reality in the curricula. Indeed, many simulators, especially for laparoscopy have been developed [1], to help surgeons in their training. These robots record tool kinematics data during training sessions that can be processed to provide meaningful feedback to a trainee. At the moment, only statistics on achievements are given such as the change in time taken to complete a task, the total path length, the error counts, the differential number of movements of each hand or a global scoring. Even if they inform on the overall level, they do not provide information on mistakes. Such a feedback is often given naturally by an expert in the field who are not usually available.

In this article, we propose a method whose goal is to give precise feedback on the mistakes made during each task execution.

Our method is based on Neural Networks (NN) explanation. To the best of our knowledge, state of the art for NN explanation methods have only been applied for classification issues, *i.e.* trying to explain why an input image produced a classification decision in the most likely class. For regression tasks and more particularly for quality score prediction, the problem differs. Indeed, determining the important input characteristics for the decision is not relevant but determining the values of the input which would lead to an increase of the score is what we are aiming for, since it would correspond to the errors made when carrying out the gesture. Thus, our contribution is threefold. First, we adapt NN explanation to regression tasks using a gradient-based method: while a forward pass leads to the score of the realization, a backward one, with a specific loss function, gives feedback on errors. Secondly, as such gradients similar to sensitivity maps are very noisy, we present a new approach based on several NN trainings, which takes advantage of their difference. This allows for gradients and therefore feedback more robust. Thirdly, we validate the approach on a synthetic database we created in order to have a ground truth on errors. Indeed, no ground truth on the mistakes is available in the literature. Thus, to the best of our knowledge, we propose the first study to evaluate feedbacks provided when assessing gesture quality. The proposed method, named Accurate GRAdient (AGRA), outperforms other methods on this dataset. In a last part, we apply AGRA on surgical tasks, using the dataset JIGSAWS [2] acquired with the da Vinci Surgical Robot [3].

[1]Sorbonne Universite, CNRS UMR 7222, ISIR, F-75005, Paris, France {millan, achard}@isir.upmc.fr

II. RELATED WORK

A. Evaluation and Feedback

Knowing which mistakes are made during a realization is essential and necessary for skill development. Several methods that provide feedback, are already present in the literature. Several methods are already present in the literature, based on traditional techniques. For example, Feygin *et al.* are interested in trajectory tracking [4]. By knowing the ideal trajectory, calculating the error at each instant, gives a natural feedback about errors. Candalh *et al.* [5] use the same approach to study different feedback modalities such as a visual, a tactile and a kinesthetic feedback. For dance movements, synchronization is required, so Kyan *et al.* [6] realign a novice's movement with an expert one to provide a score and a feedback. For tennis serves and karate tsuki, Morel *et al.* [7] create a template of the "*perfect*" gesture, thanks to an adaptation of the Dynamic Time Warping (DTW) algorithm and then, compare a new gesture to this template to obtain both evaluation and feedback. We propose in this article to take advantages of NN explanation techniques that can provide feedback for tasks where quality assessment is done using NN [8].

B. Neural Networks Explicability

Methods explaining NN decisions aim to find the contribution of each input characteristic to the output and thus produce attribution maps.

1) Attention Maps: Attention maps are mechanisms by which a NN weights characteristics according to their importance level. These maps allow the network to focus only on task-related areas. Unlike other methods presented below, weighting is learned at the same time as the network and improves the results. These maps were introduced by Xu *et al.* to automatically subtitle images [9].

Attention maps have already been implemented for gesture evaluation and feedback. Doughty *et al.* have set up two attention mechanisms: one focusing on erroneous moments and one focusing on "perfect" moments [10]. These two mechanisms of attention are temporal and not spatial, so wrong instants are indicated but errors type remains unknown. Similarly, Li *et al*. have developed a convolutional network with a spatial attention module [11] that determines image area that led to the evaluation. However, this area does not necessarily correspond to defects location. In both works, provided feedback was not evaluated.

2) Class Activation Maps: The main objective of the activation maps is to provide maps that highlight important regions for the NN decision process. They are estimated in a classification context and try to explain the most probable decision. Initial work has been done by Zhou *et al* [12] who propose to replace the last max-pooling layer of GoogLe-Net [13], by a global-average-pooling layer. The weighted sum of the feature maps (*i.e.* last convolutional layer) then leads to the Class Activation Mapping (CAM).

3) Gradient-Based Approaches: Intuitively, important gradient values correspond to input segments that have a strong influence on the result. Simonyan *et al.* [14] proposed sensitivity maps, by calculating the gradient of the output as a function of the input pixels in a classification task. If $S_c(\mathbf{x})$ is the output of the classification network for class c and $\mathbf{x}$ is the input image, then the sensitivity maps are defined by:

$$M_c(\mathbf{x}) = \frac{\partial S_c(\mathbf{x})}{\partial \mathbf{x}} \tag{1}$$

In practice, they are very noisy, and a first solution to decrease the noise is to modify the backpropagation algorithm [15], [16]. A second solution, proposed by Shrikumar *et al.* [17], consists in multiplying the sensitivity map by the input:

$$GradInput(\mathbf{x}) = \mathbf{x} M_c(\mathbf{x}) \tag{2}$$

Instead of calculating the gradients of the output as a function of the input $\mathbf{x}$, Sundararajan *et al.* [18] integrate the gradients along a path from a base $\mathbf{x}'$ to the input $\mathbf{x}$, for each dimension i:

$$IntGrad_i(\mathbf{x}) = (\mathbf{x}_i - \mathbf{x}'_i) \times \int_1^{\alpha=0} \frac{\partial S_c(\mathbf{x}' + \alpha(\mathbf{x} - \mathbf{x}'))}{\partial \mathbf{x}_i} d\alpha \tag{3}$$

During computation, the integral is approximated by a summation: gradients at the N points located on the straight line between the base $\mathbf{x}'$ and the input $\mathbf{x}$, are summed. Thus, if the base has a score close to zero, the integrated gradients form a sensitivity map of the prediction output, $S_c(\mathbf{x})$.

Smilkov *et al.* [19] propose to create an improved sensitivity map based on a smoothing of $S_c(\mathbf{x})$ with a Gaussian kernel (*SmoothGrad*). Since a direct computation of such a local average in a high-dimensional input space is impossible, a stochastic approximation is estimated by taking N random samples from the input neighborhood $\mathbf{x}$ and averaging the resulting sensitivity map:

$$SmoothGrad(\mathbf{x}) = \frac{1}{N} \sum_{i=1}^{N} M_c(\mathbf{x} + \mathcal{N}(0, \sigma^2)) \tag{4}$$

where $\mathcal{N}(0, \sigma^2)$ is a Gaussian noise with 0 mean and σ as standard deviation.

Attention maps do not provide joint information (spatial and temporal) required for meaningful feedback. Approaches using activation maps make it possible to determine which parts of the input were more relevant for the classification results. They are therefore not adaptable to regression problems. Indeed, activation maps on an evaluation task would not lead to information about positive or negative impact of input segments. Methods using the gradient seem to be the most suitable for our problem. However, they have all been used in a classification framework, so using them would need adjustments.

We therefore propose a new approach: instead of estimating the gradient of the output relative to the input, the gradient of the ideal output (maximum score) relative to the input is estimated. This allows us to determine how the input should be adjusted to get a better score. Moreover, we also introduce a new approach to deal with noisy gradients. Since no database proposes ground truth about errors position, we introduce a synthetic database where ground truth is known. This approach has been widely used in the literature to study complex phenomena. For instance, Vaughan *et al.* introduced synthetic data in order to develop a network that learn interpretable features [20].

III. SYNTHETIC DATABASE AND REGRESSION MODEL

In this section, we present the proposed database as well as the model used to estimate the quality of each signal.

A. Synthetic Database

Gesture quality evaluation databases usually provide only a score representing the gesture quality but no ground truth exists about the errors that were made. However, in order to evaluate and compare methods providing feedback, this ground truth is essential. Thus we create a database of synthetic signals, representing simple 2D movements, where quality evaluation and feedback ground truth are available. Each "*gesture*" is represented by 2 sinusoids (one for each dimension) with an amplitude between 1 and 3, to which a Gaussian noise of mean 0 with a standard deviation of 0.05 is added. Each signal is composed of 3 periods and sampled at 60 Hz, and the sinusoid frequency is randomly drawn between 0.5 and 3.5 Hz. Errors on these sinusoids are composed of random perturbations. Their number varies between 0 and 8. Finally their position is drawn according to an uniform law. Figure 1 shows how a gesture is generated.

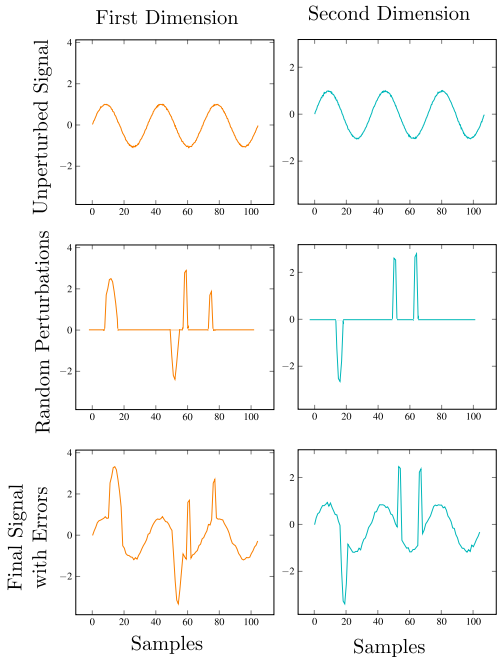


Fig. 1: Gesture generation Example. First a "perfect" signal is create, and afterwards perturbations can be added in order to generate gestures where errors were made

A score is generated by calculating the Euclidean distance between the perfect signal and the created signal. 0 is assigned to ideal signals while the score is around 10 when 8 perturbations are present. 1000 signals are generated, 750 for the training base and 250 for the test base. Two examples of signals extracted from the database are shown Figure 2.

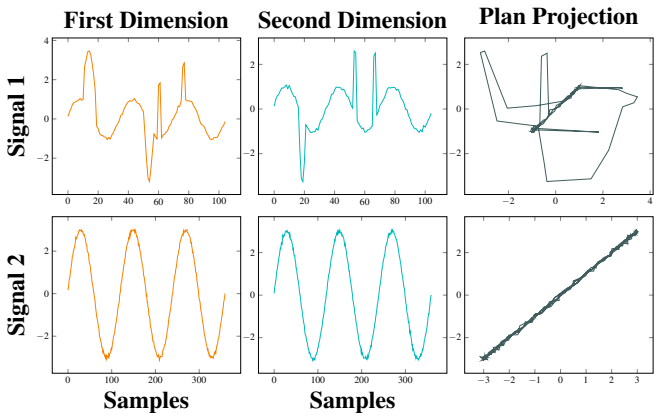


Fig. 2: Examples of two signals of the database. On top a disturbed signal with 7 errors and on the bottom an error-free signal.

B. Regression Model

The regression network consists of four layers of temporal convolution with filters (8, 8, 16, 16) of size (25, 5, 5, 5), without bias. Each of them is followed by a pooling layer of size 3. Two layers of fully connected neurons of size 50 and 1, also unbiased, end the network. The model is presented Figure 3.

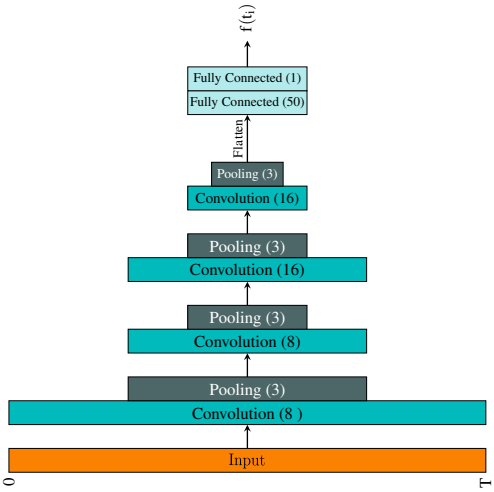


Fig. 3: Neural Network model used to regress scores on the synthetic database.

During training, a dropout rate of 0.5 is applied to the 50 neurons layer, to avoid overfitting. Training is performed using the ADAM algorithm [21] with a learning rate of 0.01, for 100 epochs. In order to obtain 50 different models, 50 trainings are performed. The average MSE of the 50 models, over the test set, is 0.619 with a standard deviation of 0.089. Thus, during prediction, these models have similar behaviors.

IV. ACCURATE GRADIENT - AGRA

As stated before, we decided to use a NN explanation method. As the aim of the method is to evaluate the quality of a gesture, we are thus in a regression context and not a classification one, which requires a change in the paradigm of current state of the art methods. The direct application of these methods leads to calculate the gradient of the output with respect to the input. However, this gradient is not very informative since it indicates the moments of the signal that have a strong importance on score regression. To provide useful feedback, we propose to change the loss function used to compute the feedback, so it will not be the same that during the learning step. Actually, it must represent the difference between a perfect gesture and the studied gesture. Thus, the new cost function used is:

$$l_2(\mathbf{x}) = (\hat{s}(\mathbf{x}) - score_{max})^2 \tag{5}$$

where $\hat{s}(\mathbf{x})$ is the prediction of the network and $score_{max}$, the score obtained by error-less gestures. Gradients obtained using the cost function l_2 highlight moments when errors have been made, but the amplitude remains very far from the ground truth. Moreover, all gradients are very noisy and all the erroneous moments are not highlighted, as illustrated by **Grad** plots Figure 4.

In order to better recognize errors, a second modification is proposed. It consists in training the input $\mathbf{x}$, with the l_2 loss, so that it obtains the best score according to the model.

Algorithm 1 Feedback Computation

Require: $\mathbf{x}, \lambda, \varepsilon$
Ensure: $GRAD(x)$
  $\mathbf{x}' = \mathbf{x}$
  while $l_2(\mathbf{x}') > \varepsilon$ **do**
    $grad = \frac{\partial l_2(\mathbf{x}')}{\partial \mathbf{x}'}$
    $\mathbf{x}' \leftarrow \mathbf{x}' - \lambda grad$
  end while
  $Grad(x) = \mathbf{x} - \mathbf{x}'$

The "*correction*" of the input is done following Algorithm 1, where λ is the learning rate and ε is the tolerance: the loop stops when the difference between the maximum score and $\hat{s}(\mathbf{x})$ is less than ε. As mentioned before, this gradient is very noisy [19], [22]. Moreover, during the experiments, we observed that it depends strongly on weights initialization and on network training. Thus, even if two different trainings lead to similar regression scores, feedbacks are highly variable. We decided to take advantage of these feedbacks variations. Indeed, if two models can highlight different moments with errors, then by training a large number of models, it is possible to find all the errors. The proposed method, named *Accurate GRAdient* (*AGRA*), is based on this assumption. The main idea is to train N models M_i on the same quality evaluation task, but with different initializations of the model weights. Once these N models have been trained, averaging gradients obtained by each model gives a feedback highlighting all the errors. This method is expensive in terms of computing time since it requires training N networks on the same task. However, gradients have a strong dependence on network initialization, which justifies training several networks to provide meaningful and understandable feedback.

V. EXPERIMENTAL RESULTS

The method presented Section IV is first tested on the synthetic dataset presented Section III-A. To determine its contribution, we compare it with other state of the art methods. In the following section, we first present qualitative then quantitative results of the different methods, For all the methods involved in this section, the loss function $l_2(x)$ previously defined is used to compute gradients. Results for the following five methods are presented afterwards:

- Gradient *Grad* [14] calculated with the algorithm 1, a learning rate, λ of 0.1 and a tolerance, ε of 0.015.
- *Grad*×*Input* as defined in the equation 2 and proposed by [17], [23].
- *SmoothGrad* [19] estimated as the mean of 50 gradients obtained with the Algorithm 1 by adding a Gaussian noise of mean 0 and standard deviation 0.1 on the input signal (equation 4).
- *IntGrad* [18]. Since the proposed network has no bias, the $\mathbf{x}'$ base is set to a zero signal of the same length as $\mathbf{x}$. Under these conditions, the score of the base is $\hat{s}(\mathbf{x}') = 0$ and the method *IntGrad* can be interpreted as a sensitivity map of the prediction output $\hat{s}(x)$.
- The AGRA method with 50 models, with a learning rate λ of 0.1 and a tolerance ε of 0.015.

For all methods, which do not involve averaging across multiple models, a random model was chosen from all models, and remains the same for all methods and results presented.

A. Qualitative Results

As shown Figure 4, the (*Grad*) method is very noisy and does not give clear and easily interpretable results. Multiplying this noisy gradient by the input only amplifies the noise and makes the results even less interpretable. Interesting peaks are more distinct, but the overall results seem more noisy than before. In addition, the gradient sign which gives information about error direction, is lost because of this multiplication. Using the *SmoothGrad* method gives better qualitative results. However, noise is still present and results are again difficult to interpret. In addition, gradient amplitude on errors is often smaller than the ground truth. The *IntGrad* method gives very noisy gradients, which show peaks at undisturbed locations, making it very difficult to interpret. The least noisy and most accurate results are obtained with the AGRA method. The method effectively highlights the samples corresponding to the perturbations, with the right sign, leading to a clear and easily interpretable feedback.

Moreover, it is possible to reconstruct what the network considers to be an ideal signal, by adding the computed gradient to the initial signal. In the ideal case, these reconstructed signals are assumed to be a line, like the second signal Figure 2.
We compare AGRA and *SmoothGrad* (that seems to be the two most promising methods) on two signals Figure 5 and find that AGRA method performs the best reconstruction.

B. Quantitative Results

To compare methods more thoroughly, we also computed quantitative results. As the ground truth is available for each example, it is possible to compute the ideal feedback (the difference between the disturbed and ideal signals) and compare it with the results obtained with the different methods. Two measurements are used to make this comparison:

- Mean Square Error (MSE) between the error-free signal and the reconstructed signal obtained from the gradients. This metric cannot be used for methods such as *Grad*×*Input* or *IntGrad*, because their purpose is only to highlight important temporal steps and not to reconstruct a perfect signal.
- The Pearson correlation coefficient between the ideal gradient and the gradient obtained with the different methods. To avoid penalizing methods that do not handle signs (*Grad*×*Input* and *IntGrad*), this coefficient is calculated between the norms of the ideal gradient and the gradient obtained with the methods.

The 250 test examples were used for these two metrics and the results presented are the average over these examples.

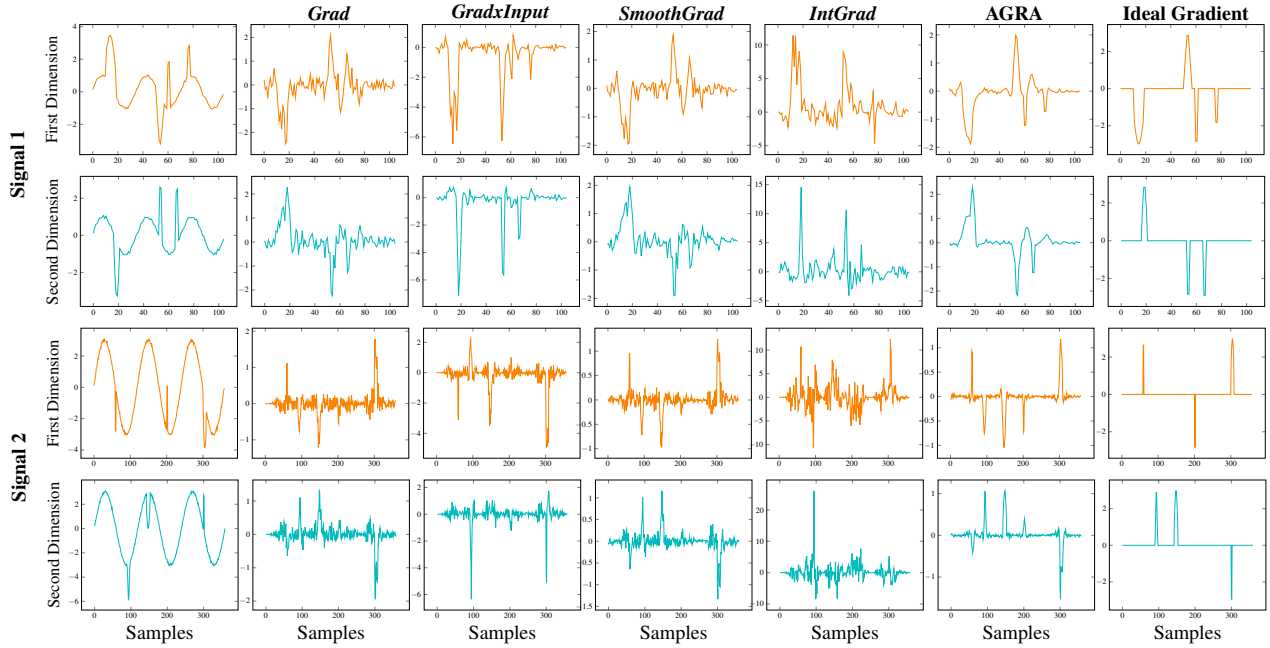


Fig. 4: Qualitative results of all methods for 2 signals with different errors.

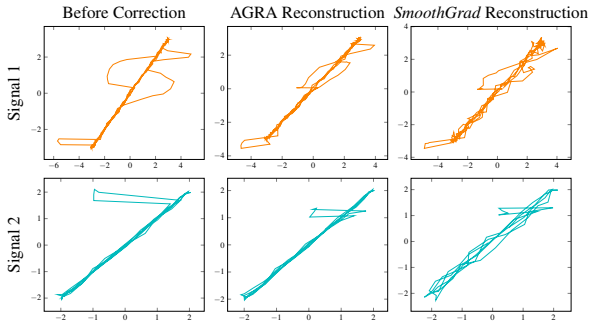


Fig. 5: Correction obtained with the AGRA and *SmoothGrad* methods applied on a test signals.

Moreover, for the methods *Grad*, *Grad×Input*, *SmoothGrad* and *IntGrad*, the calculation of metrics was done on the 50 models and then averaged.

Methods	MSE	Pearson Correlation (Norm)
Grad [14]	5.81	0.85
GradxInput [17], [23]	**NA**	0.86
SmoothGrad [19]	6.10	0.84
IntGrad [18]	**NA**	0.67
AGRA	**5.39**	**0.95**

TABLE I: Results for the two proposed metrics.

As a reminder, a perfect feedback would lead to a MSE of 0. Table I presents the MSE and Pearson's correlation obtained with different methods. Both *Grad* and *SmoothGrad* gradient are noisy, leading to bigger MSE than AGRA method. Concerning defaults detection, the Pearson correlation is a better indicator, and can be computed for all methods. Best results are obtained with the proposed method, which confirms the previous qualitative study and proves that this method gives better results than the state of the art ones.

To study the behavior of the AGRA method, it is relevant to show the evolution of Pearson correlation and MSE, as a function of the number of averaged models (Figure 6).

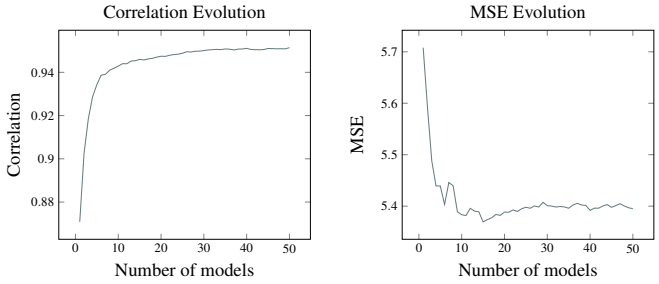


Fig. 6: Evolution of MSE and correlation as a function of the number of averaged models.

Overall, correlation and MSE improve over iterations and stabilize around 50. Remember that the different models differ only by their weights initialization. Regression scores are therefore the same, but gradients differ greatly. It is impossible to *a priori* define models that will lead to useful feedback. By averaging gradients obtained by 50 or more models, provided feedbacks are understandable regardless of network initialization. The same reasoning can be applied to the Pearson correlation coefficient.

VI. FEEDBACK FOR SURGERY

The AGRA method shows the best qualitative and quantitative results so the next step consists in applying it on a

real dataset. We chose the JIGSAWS dataset [2], which is composed of different surgical tasks done with the da Vinci surgical robot.

A. Model presentation

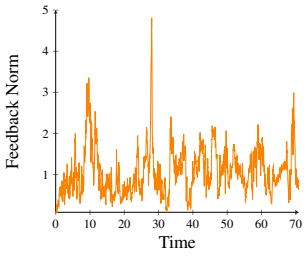
To extract meaningful features for gesture quality evaluation, temporal convolution are used. Inputs are composed of kinematic data which include position, rotation matrix, linear and rotational velocity, as well as gripper angle velocity, for both slave and master parts, leading to a descriptor of size $(T \times 76)$. All signals are padded with zero, so that they all have the same input length. The model embedded in this architecture is composed of 4 temporal convolutional layers: the first two with 9 filters of size 5, and the last two with 18 filters also of size 5. Each one is followed by a max-pooling layer of size 5. To ensure a global score, features maps are flattened after the last max pooling layer in order to get a vector compatible with fully connected layers. Afterwards, two fully connected layers are added of size 100 and 1, in order to predict the global score. The L2 regularization is used on the weights of each layer with a coefficient of 0.1 to limit overfitting. Adam algorithm [21] is used for the retropropagation using an initial learning rate of 0.001 and a batch size of 4. Just like for the synthetic dataset, scores inverse is predicted: perfect gestures get a score of 0 and really bad ones get a score of 30. 10 networks are trained to provide feedback with the AGRA Methods.

B. Results

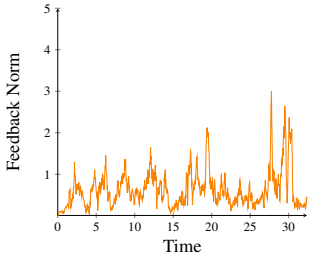
MSE results are presented in Table II. As scores are evaluated on a scale going from 5 to 30, obtained MSE are quite good. Since scores need to be perfectly regressed, results in the following are computed using instances from the Knot Tying task.

	Suturing	**Needle Passing**	**Knot Tying**
Regression	4.40 ± 0.25	3.79 ± 0.12	2.89 ± 0.21

TABLE II: MSE results for score prediction.



(a) Feedback norm for an instance with a true score of 9, and an estimated score of 9.25 ± 1.51.



(b) Feedback norm for an instance with a true score of 22, and an estimated score of 20.17 ± 0.69.

Fig. 7: Illustration of the provided feedback for two Knot Tying tasks with different scores.

Kinematic data are composed of 76 dimensions at any given time and therefore the feedback provided by AGRA is also composed of 76 dimensions. It is difficult, from this multi-dimensional signal of variable length, to follow up the learner's skills. Thus, we estimate AGRA gradient norm at every moment to highlight moments considered erroneous by the model as shown Figure 7a and 7b for two gesture realisations. These results are coherent since the feedback has less peaks when the score is closer to the perfect score (30). Furthermore, to check the meaningfulness of the feedback, we studied the correlation between real scores and sum of the gradient norm along the temporal dimension. When scores are high, this sum is supposed to be small and vice versa.

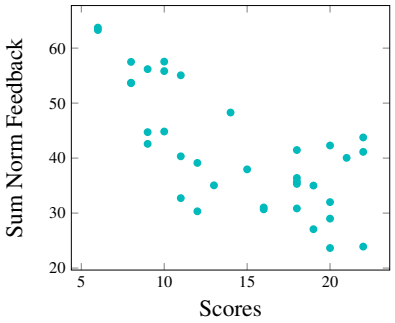


Fig. 8: Norm feedback sum according to real scores.

Figure 6 shows that it is indeed the case, with a Pearson Correlation of -0.72. Without ground truth on the errors, it is difficult to go further in the interpretation of feedback quality, but this high correlation value is promising.

VII. CONCLUSION

In this article, we proposed a method to provide automatic feedback after gesture evaluation.
Among all methods performing NN explanation, we have chosen to use sensitivity maps to obtain this feedback. Several adaptations have been done to work on regression tasks. Even with these adaptations, sensitivity maps are very noisy, not precise and lead to a feedback that is difficult to understand. Moreover, it strongly depends on the network weights initialization. We decided to take advantage of these variations to average the gradients coming from different learning processes and obtain a robust gradient, able to detect all errors. In order to test this approach, a synthetic database, composed of 2D signals, has been created. It has the advantage of being associated with a ground truth where signals defects are known. Comparing the proposed method, AGRA, with state of the art ones, it appears that AGRA gives better results both qualitatively and quantitatively. Furthermore, when tested on robotic tasks, AGRA shows a correlation between the feedback norm and the real score. This result is encouraging and we propose, afterwards, to create an annotated database with a score but also errors in order to validate this approach on real data and, most importantly, to be able to create a real virtual coach.

REFERENCES

[1] M. Alaker, G. R. Wynn, and T. Arulampalam, "Virtual reality training in laparoscopic surgery: a systematic review & meta-analysis," *International Journal of Surgery*, vol. 29, pp. 85–94, 2016.

[2] Y. Gao, S. S. Vedula, C. E. Reiley, N. Ahmidi, B. Varadarajan, H. C. Lin, L. Tao, L. Zappella, B. Bejar, D. D. Yuh, C. C. G. Chen, R. Vidal, S. Khudanpur, and G. D. Hager, "JHU-ISI Gesture and Skill Assessment Working Set (JIGSAWS): A Surgical Activity Dataset for Human Motion Modeling," *Modeling and Monitoring of Computer Assisted Interventions (M2CAI) – MICCAI Workshop*, p. 10, 2014.

[3] G. Guthart and J. Salisbury, "The intuitive/sup tm/ telesurgery system: overview and application," *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No.00CH37065)*, vol. 1, pp. 618–621 vol.1, 2000.

[4] D. Feygin, M. Keehner, and F. Tendick, "Haptic guidance: experimental evaluation of a haptic training method for a perceptual motor skill," *Proceedings 10th Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems. HAPTICS 2002*, pp. 40–47, 2002.

[5] N. Candalh-Touta and J. Szewczyk, "How can we improve the training of laparoscopic surgery thanks to the knowledge in robotics ?" *15th International Conference on Education and Information Systems, Technologies and Applications*, 2017.

[6] M. J. Kyan, G. Sun, H. Li, L. Zhong, P. Muneesawang, N. Dong, B. Elder, and L. Guan, "An approach to ballet dance training through ms kinect and visualization in a cave virtual reality environment," *ACM Trans. Intell. Syst. Technol.*, vol. 6, pp. 23:1–23:37, 2015.

[7] M. Morel, C. Achard, R. Kulpa, and S. Dubuisson, "Automatic evaluation of sports motion: A generic computation of spatial and temporal errors," *Image Vis. Comput.*, vol. 64, pp. 67–78, 2017.

[8] M. Millan and C. Achard, "Fine-tuning siamese networks to assess sport gestures quality," in *Proceedings of the 15th International Conference on Computer Vision Theory and Applications*, 2020.

[9] K. Xu, J. Ba, R. Kiros, K. Cho, A. C. Courville, R. Salakhutdinov, R. Zemel, and Y. Bengio, "Show, attend and tell: Neural image caption generation with visual attention," *ICML*, 2015.

[10] H. Doughty, W. Mayol-Cuevas, and D. Damen, "The pros and cons: Rank-aware temporal attention for skill determination in long videos," *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7854–7863, 2019.

[11] Z. Li, Y. Huang, M. Cai, and Y. Sato, "Manipulation-skill assessment from videos with spatial attention network," *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, pp. 4385–4395, 2019.

[12] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, "Learning deep features for discriminative localization," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2921–2929.

[13] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1–9, 2015.

[14] K. Simonyan, A. Vedaldi, and A. Zisserman, "Deep inside convolutional networks: Visualising image classification models and saliency maps," *CoRR*, vol. abs/1312.6034, 2013.

[15] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *European conference on computer vision*. Springer, 2014, pp. 818–833.

[16] J. T. Springenberg, A. Dosovitskiy, T. Brox, and M. A. Riedmiller, "Striving for simplicity: The all convolutional net," *CoRR*, vol. abs/1412.6806, 2015.

[17] A. Shrikumar, P. Greenside, and A. Kundaje, "Not just a black box: Learning important features through propagating activation differences," in *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. JMLR. org, 2017, pp. 3145–3153.

[18] M. Sundararajan, A. Taly, and Q. Yan, "Axiomatic attribution for deep networks," in *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. JMLR. org, 2017, pp. 3319–3328.

[19] D. Smilkov, N. Thorat, B. Kim, F. Viégas, and M. Wattenberg, "Smoothgrad: removing noise by adding noise," *arXiv preprint arXiv:1706.03825*, 2017.

[20] J. Vaughan, A. Sudjianto, E. Brahimi, J. J. Chen, and V. Nair, "Explainable neural networks based on additive index models," *ArXiv*, vol. abs/1806.01933, 2018.

[21] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *CoRR*, vol. abs/1412.6980, 2015.

[22] B. Kim, J. Seo, S. Jeon, J. Koo, J. Choe, and T. Jeon, "Why are saliency maps noisy? cause of and solution to noisy saliency maps," *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, pp. 4149–4157, 2019.

[23] S. Bach, A. Binder, G. Montavon, F. Klauschen, K.-R. Müller, and W. Samek, "On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation," *PloS one*, vol. 10, no. 7, 2015.