## A. MODEL SUMMARY

* Competition Name: Predicting Molecular Properties
* Team Name: DL Guys
* Private Leaderboard Score: -3.14969
* Private Leaderboard Place: 5
* Team members

|  |  |  |
| --- | --- | --- |
| Name | Location | Email |
| Guillaume HUARD | Paris, France | grj.huard@gmail.com |
| Thanh Tu NGUYEN | Paris, France | thanhtu19392@gmail.com |
| Tung Lam DANG | Paris, France | tunglam.dang@gmail.com |

### 

### A2. Background on you/your team

If part of a team, please answer these questions for each team member. For larger teams (3+), please give shorter responses.

1. Guillaume HUARD
   * What is your academic/professional background: Msc Data Science, Data Scientist at BNP Paribas Cardif
   * Did you have any prior experience that helped you succeed in this competition: No prior knowledge on GNN or quantum chemistry
   * What made you decide to enter this competition: Interesting problem and opportunity to discover GNN
   * How much time did you spend on the competition: Average 20h/week for 3 months
   * If you competed as part of a team, who did what: I contributed 14 predictions and 30 checkpoints of 14 GNN as part of the final stacking
2. Thanh Tu NGUYEN
   * Msc Financial Engineering, Data Scientist at BNP Paribas Cardif
   * Did you have any prior experience that helped you succeed in this competition: No prior knowledge on GNN or quantum chemistry
   * What made you decide to enter this competition: Interesting problem, gain experience in Kaggle competitions and talking with my teammates
   * How much time did you spend on the competition: Average 15h/week in 2 last months of competition
   * If part of a team, how did you decide to team up: to get interesting ideas and insights from my teammates about this competition.
   * If you competed as part of a team, who did what: I contributed 3 predictions of 3 different GNN as part of the final stacking
3. Tung Lam DANG

* MSc Statistics & Econometrics
* Data scientist at BNP Paribas Cardif
* No prior knowledge on GNN or quantum chemistry
* Experience is Deep Learning and Kaggle competitions
* What made you decide to enter this competition: Interesting problem and opportunity to discover GNN
* How much time did you spend on the competition: Average 10h/week for 3 months
* I contributed 3 predictions of 3 GNN as part of the final stacking

### A3. Summary

* On macro level our best submission is a 2 layer stacking:
* The base level consists of different variant of the general Graph Neural Net with edge, node and global representation with some variations
* It was implemented with pytorch and pytorch\_geometric.
* The 2nd level is some metamodel trained on our validation set of 5000 molecules : 1 linear stacking model and 1 LGBM (cf. Stacking section below)
* The final submission is a blend of 2 meta model

## Stacking:

* Our single best model is the one with edge to edge convolution which gives us -2.9. But we have various models around -2.7 which are variants of it. By stacking all (20 models) we got -3.13 on LB.
* Another thing we found out at the last day helps improve our score from -3.13 to - 3.15 is adding checkpoints of our models to stacking pipeline. So finally, we have 50 predictions to do stacking.

Our final result is a blend of LGBM and HuberRegressor.

* LGBM: 20 GroupKFold on all bond types together
* HuberRegressor: 20 GroupKFold on every bond type separately.

### A4. Features Selection / Engineering

## Architecture:

The final architecture is based on the paper <https://arxiv.org/abs/1812.05055>.

We tried different variations to improve this architecture, here is a summary of what worked and what didn’t work:

* Normalization: We found that LayerNorm worked better that BatchNorm for this data and helps improve convergence
* Softplus vs ReLU: Softplus did provide a ~ 0.1 boost of logMAE for our models vs a ReLU baseline
* Edge to node message gating: We found that adding some gating mechanism to the edge representation before the scatter*mean (see torch*geomrtric) for node update helps
* Edge to edge convolution: Guillaume implemented something that seem to work very well. He noticed after a feature importance test that the most important one was by far was the angle between an edge and the edge with the closest atom to the first edge. To integrate this angle feature for more than the closest edge, we updated each edge with a convolution of the edge in question and its neighboring edges in the graph (more specifically the neighboring edges that chemically connects two atoms), and putting in this convolution the angle of the edge vectors. This architecture tweek made our architecture 5 times slower but gave us a 0.15 improvement compared to the best model without it.
* 1 prediction tail per type: All types share a GNN “body”, but we found that having different MLP for each type helps.
* In some variants, before feeding into output MLP layers, we pool all the edges and nodes in the chemical bond path from atom*0 to atom*1. It seems to have helped in the beginning of the competition but our best model did not use it.
* For our architectures, we found that having a representation of the link between atom*0 and atom*1 is important. Also including the global representation as inputs of the top layers is important

### A5. Training Method(s)

* Cf. A3

### A6. Interesting findings

* Cf. A4

### A7. Simple Features and Methods

* Is there a subset of features that would get 90-95% of your final performance? Which features? No, there is not particular features. We just found out that the most important one was by far was the angle between an edge and the edge with the closest atom to the first edge.
* What model that was most important? KNN model (cf. A4)

### A8. Model Execution Time

Many customers care about how long the winning models take to train and generate predictions:

* How long does it take to train your model?
  + Lam: 100h
  + Thanhtu: 50h
  + Guillaume: 500h
* How long does it take to generate predictions using your model?
  + Lam: 30 min
  + Thanhtu: 7 min