We thank the reviewers for their insightful and positive feedback. We are encouraged they found our motivation and idea to be clear (R4), new (R2, R3), and novel (R3), well organized and clarified w.r.t. prior works (R3, R4), and our theoretical analysis reliable/well-supported (R3, R4). We are glad they found our method to be intuitive (R3, R4), evaluated with extensive/convincing/carefully worked experiments (R2, R3), and achieving consistent significant/promising improvements (R2, R3, R4). We address reviewers comments below and will incorporate all feedback in the final version.

Q. [R1] "Over explored topic, and "okay" results."
A. We disagree with this statement. (1) The geometric learning on graph has attracted an increasing attention, because it is an important topic; but still it has many unexplored questions such as interaction learning, where this paper tries to work out. (2) Our method achieves SOTA results w.r.t. baselines, evidenced by averaged performance with standard deviation, and the further ablation study and visualization show significant improvements, which are endorsed by other three reviewers.

Q. [R1]."What justifies so many transformations and back and forth between spaces?"
A. In order to bridge the geometry gap for interactive learning and maintain conformal invariance of each space, we utilize the *exponential* and *logarithmic* mappings to transform features between spaces.

Q. [R1] "It would ... interesting to make a stronger case ... by adding running time, or other performance metrics"
A. The time complexity is analyzed quantitatively in Section 3.3, which is more exact and computationally efficient. The comment about the time consumption of basic operations in different spaces is reasonable, where operations in hyperbolic space indeed consume more time than Euclidean space. For instance, GAT and HGAT cost 3.48s and 10.59s respectively on Cora in node classification. The extra time consumed is also within an acceptable range and doesn't affect it's extendibility.

Q. [R1] "Few experiments, it would be interesting to try out other task, such as graph classification."
A. We disagree with "few experiments". Convincing/extensive experiments are endorsed by other reviewers. Besides, our approach is to endow each node the freedom to determine the importance of each geometry space, therefore, the node-level tasks are sufficient to validate the effectiveness of GIL, including node classification and link prediction.

Q. [R1] "Lack of code." [R2] "Will the authors publish the full code and evaluation settings ... ?"
A. As we cannot report an external link here during rebuttal, we will open the source code after blind review. For evaluation settings, we closely follow the parameter settings in HGCN mentioned in Section 4.1. The detailed settings are supplemented as follow. Model configuration: Except for shallow methods with 0 layer, other methods use the same 2 hidden layers. Training configuration: All methods use the following training strategy, including the same random seeds for initialization, and the same early stopping on validation set with 100 patience epochs. We measure performance on the test set over 10 random parameter initializations. The optimal L_2 regularization with weight decay [1e-4, 5e-4, 1e-3] and dropout rate [0.0-0.6] are obtained by grid search for each method. We will add the details and release our source code in the final version.

Q. [R2] "How does the model compare against ... constant curvature spaces with fixed or learnable curvatures, e.g.[2]?"
A. The constant curvature spaces are a unified form of different curvature spaces. The difference between proposed method and the latest works about constant curvature spaces, is that these methods still embed the entire graph in a space of uniform curvature, although the curvature is fixed or learnable. While our method give each node privilege to determine proper features from different curvature spaces, specifically using hyperbolic and Euclidean space here. More importantly, there is no interaction learning between different spaces in existing works. Thanks for pointing the newly related work though it was published after NeurIPS submission deadline. We will add more discussion in the final version.

Q. [R2] ...the problem of permutation variance –>"...Can the authors detail this statement?"
A. Hyperbolic GNNs suffer from the problem of permutation invariance, because the basic operations in hyperbolic space such as addition cannot hold the property of commutative and associative, it cannot maintain the permutation invariance if apply GNN operate directly in hyperbolic space. Therefore, transfer operations such as aggregation into tangent space is a intrinsic choice to deal with this problem.

Q. [R2] "why does it preserve properties and structures of the original space?"
A. Because of different closed-form operations in each space, the feature interaction learning first transform different features into its own space and then fuse them, therefore, it can maintain conformal invariance of each space.

Q. [R4] "What if the two models are trained offline, before being combined by a weight hyperparameter?"
A. The suggested method was added as a baseline EucHyp in the paper. The results show that our algorithm significantly outperform this method in Table 2.