A Appendix

A.1 Code Availability

The code used for this study is available at https://github.com/tum-ai/reaction_conditions.

A.2 Data Preprocessing

We derive the training dataset from [17]. After identifying unique reactions based on the materials involved, we merge instances where the same materials in precursors and target material are reported by calculated the average sintering temperature. Data points that deviated by more than two standard deviations are excluded. Additionally, all temperatures in the dataset were filtered for sintering temperatures and converted to Celsius, retaining only those above 400°C. This ensures plausibility of the respective experimental parameters.

A.2.1 Example Reactions

Note that our model does not utilize information about reaction coefficients or by-products. By-product information is excluded to enable the model's application in proposing synthesis parameters within a materials discovery operation. Below two example reactions from the Kononova dataset are shown [17]:

1. Reaction:

$$4\,\mathrm{BaCO_3} + 3\,\mathrm{Nb_2O_5} + \mathrm{Nd_2O_3} + 4\,\mathrm{ZrO_2} \rightarrow \mathrm{Ba_4Nd_2Zr_4Nb_6O_{30}} + 4\,\mathrm{CO_2}$$

2. Reaction:

$$\mathrm{Na_3PO_4 \cdot 12H_2O} + 0.16\,\mathrm{Nb_2O_5} + 2\,\mathrm{SiO_2} + 1.6\,\mathrm{ZrO_2} \rightarrow \mathrm{Na_3Zr_{1.6}Nb_{0.8}Si_2PO_{12}} + 12\,\mathrm{H_2O}$$

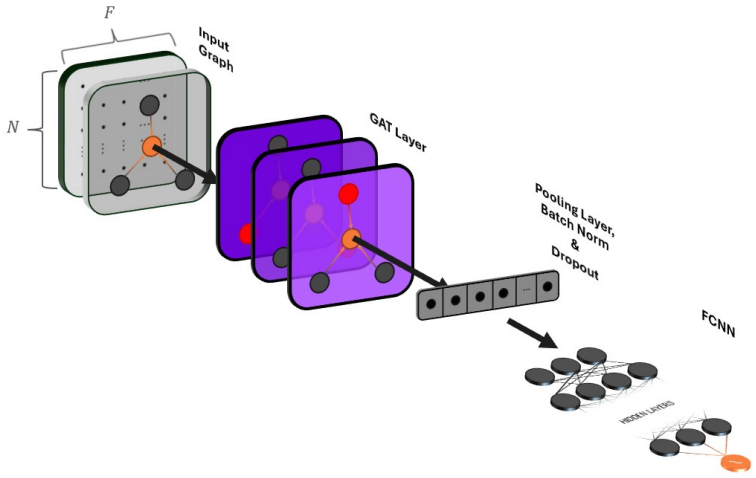

Figure 3: A detailed overview of the RGN architecture, which models inorganic reactions and performs point regression to predict reaction conditions. In this model, N represents the number of compounds, and F denotes the number of features. The architecture leverages a series of GAT layers to capture node-level attention, followed by pooling layers and batch normalization to refine feature representations, ultimately feeding into a fully connected neural network for final predictions.

A.3 Hyperparameters Tuning

We present the best hyperparameters found for various setups used in our experiments. For each model variant, we conducted 20 trials per fold, optimizing performance across multiple configurations. The table below summarizes the best hyperparameters identified during this study.

Table 2: Best hyperparameters found in the hyperparameter study.

Study	Epochs	Base Layer Dim (log)	Batch Size (log)	NN Layers	GAT Steps
mtencoder_512	50	10	6	4	4
mtencoder_256	33	10	6	4	4
composition	39	9	4	4	2
matscibert	39	9	4	4	2
matminer	33	10	4	2	3

Hyperparameter Description:

- **Epochs**: The total number of training iterations over the entire train dataset.
- **Base Layer Dim (log)**: The size of the base layer (input layer) of the neural network head, expressed in logarithmic scale.
- **Batch Size (log)**: The number of samples processed in a single forward and backward pass. Smaller batch sizes lead to more frequent updates.
- **NN Layers**: The total number of neural network layers. This parameter defines the depth of the model, with deeper architectures able to capture more complex relationships in the data.
- **N Layers**: The number of fully connected or specific neural network layers. This influences the capacity of the model to learn hierarchical representations from the input data.
- **Conv Steps**: The number of GAT layers in the model. This controls how many times the convolution operation is applied and to which extend information can be exchanged in the graph.

A.4 RGN Architecture Ablation

We investigate different architectures of the employed graph neural network for reaction condition regression. For each architecture we conduct a hyperparameter study as mentioned above. For message passing we use GAT (results in table 3).

Graph Structures: Describes the edge types in the graph, where we distinguish between two connection types: *all-to-all* and *precursor-to-target*. *All-to-all* implies that every node connects to every other node, offering greater expressiveness to the GNN. *Precursor-to-target* means that only precursor nodes connect directly to the target node, imposing a structural constraint. Interestingly, we found this constraint advantageous, as it enabled the GNN to capture more relevant, context-specific information and effectively summarize essential interactions.

Pooling Graph Level Representations: This study examines two pooling operations: *mean pooling*, which produces a more homogeneous message by averaging node features, and *add pooling*, which accounts for graph size and can amplify prominent node features. pooling demonstrates superior performance, likely due to its capacity to preserve the graph's size and the quantity of precursors.

Connection Type: We examined two versions of edge connections between nodes: *undirected* and *directed*. The undirected case was tested with both all-to-all and precursor-to target structures, while the directed case was found to be more beneficial when used only with the precursor-to-target structure. The undirected form requires a symmetric adjacency matrix, which introduces more parameters, whereas the directed precursor-to-target structure reduces the parameter count. This reduction in parameters was advantageous, as it simplified the model, helping to prevent overfitting and enhancing the GNN's ability to focus on essential, directional interactions.

A.5 Parity Plot of Results

Table 3: Ablation study results for different GNN configurations on reaction condition regression. The values represent test mean absolute errors (MAE) in °C for each configuration.

Configuration	**Connection Type**	**MAE**
All-to-All	Undirected	97.7
Prec to Target	Undirected	100.4
Prec to Target	Directed	91.6
Prec to Target **+ add pooling**	Directed	**89.9**

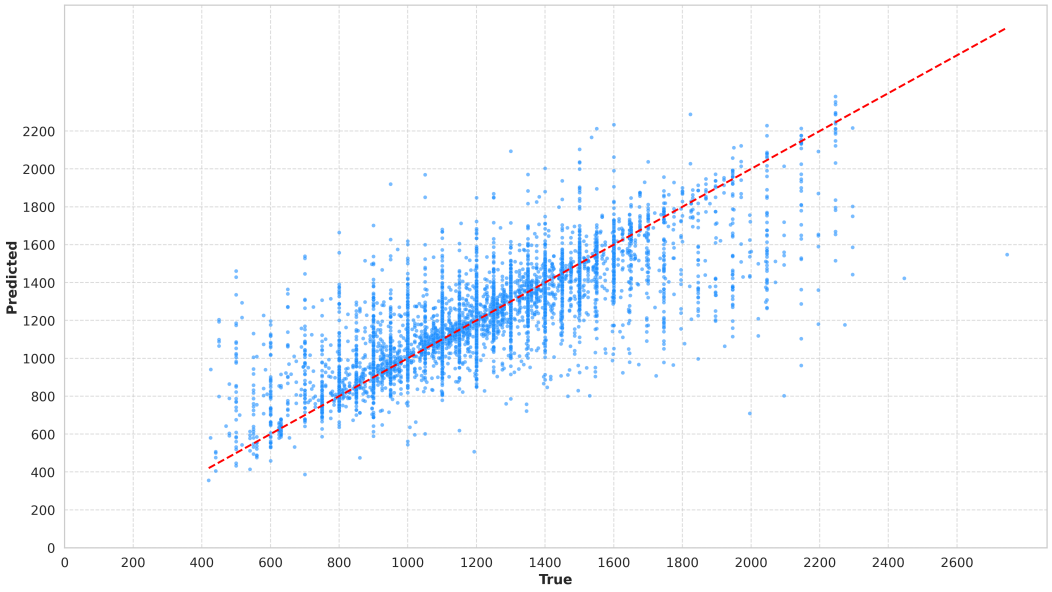

Figure 4: We present the true versus predicted sintering temperatures for test data points across all five folds, covering the entire dataset, using the best performing model with MTEncoder 512.