P0: Thanks for having me at XXX.

P1: I'd like to take two minutes to talk a little more about myself. My background is in data mining. My research is to discover knowledge from graph and text data. During my PhD, I worked on social recommender systems. Basically it was predicting user interactions on social networks. When I was at Carnegie Mellon, I worked on graph anomaly detection. The task was detecting fake accounts on Twitter. And when I was a postdoc in UIUC, I worked on building knowledge graphs from text. Most of my work was published in KDD and I have attended the conference for ten years. Here KDD is for Knowledge Discovery and Data Mining.

P2: To simply say, knowledge discovery is mining useful and surprising patterns from big messy data. And people develop graph and text data mining techniques.

[click] Now we know Generative AI is a hot topic. That's because we are amazed and surprised by so many things from it. For example, ChatGPT can answer your questions, can write stories, and Sora can generate a one minute video from text.

[click] While I believe Generative AI should appreciate the availability of big data and also the data processing methods, these generative models are now ready to make their generated small data useful and even impactful.

[click] Since I joined the faculty at Notre Dame, I have been working on graph and text data generation for broader impacts. Today I'll introduce some interdisciplinary research projects in my lab.

P3: Before we talk about concrete things, I'd like to say, the research in this talk is kind of diverse. Due to the time limit, it has limited detail. So please feel free to ask questions. You can find paper links at the bottom of slides. Slide numbers are at the bottom right. And I want to say, this talk would not be possible without my collaborators, and amazing students. You can find the leading authors at the top right of slides.

P4: Our lab is working on at least four areas. They are material science, online education, security, and mental health. Today I will mainly talk about AI for material discovery.

P5: Here we are interested in polymer materials. Polymers have many types of properties. For example, if the permeability of carbon dioxide is very high and the permeability of methane is relatively very low, then we can make a membrane to efficiently separate natural gas.

Let's look at these data points. The horizontal axis is carbon dioxide permeability, and the vertical axis is the selectivity over methane. So, the ideal material should have very high permeability and very high selectivity.

Besides the gas industry, these polymers are also used to make face masks for your physical health in polluted places.

P6: So, what is going on in polymer research?

[click] It is very expensive to measure the polymer properties in wet labs. They spent over 60 years and collected about 700 data points. Look at these red points. In 1991, they defined an upper bound to say, if you found a new polymer above this bound, it could be a novel and useful material.

[click] Then people found these blue points called TR polymers, and they refined the upper bound in 2008.

What do these data points look like? Yeah, they are graphs. The nodes are atoms and the edges are chemical bonds. Here we want to predict their permeability and selectivity. These properties are not categorical labels. They are continuous values.

P7: So, if we use the terminology of machine learning, this task is graph regression. We can come up with at least three types of machine learning ideas to address some challenges. First, suppose we do cross validation, we will have only 500 labeled examples for supervised learning. So the models will overfit the training data.

Second, the label distribution is imbalanced. We have very few data points in this high permeability area. So the predictions are usually inaccurate. But actually this is the most interesting and important area, because we want to discover novel polymers.

And third, we don't have many labeled points, but we have millions of unlabeled points. Also, we have at least six types of gas. So we should figure out how to transfer the knowledge from unlabeled data and from different domains.

[click] Now you can see, when we apply machine learning for material science, the key problem is actually on data!

P8: In the next **20 minutes**, I will talk about a specific technique, called Data Augmentation. We will see how to use it for polymer property prediction. Well, Wikipedia says, data augmentation is to increase the amount of data by slightly modifying existing data or creating synthetic data.

[click] It has been widely used in supervised learning and self-supervised learning. In supervised learning, given us data point x and label y, we can modify x slightly to be x' and assume the label of x' is still y. Then we can train the representation learning function f and prediction function g with more data to reduce overfitting.

In self-supervised learning, we can modify x into two different examples. We assume their representations should be very similar, and we pre-train the functions without labels.

P9: Data augmentation had great success in computer vision and NLP. You know, we can flip or rotate an image. We can replace a word with its synonym. And the label should be the same, having a dog or positive review.These heuristics were effective on image and text classification.

[click] Then, how about graph data? On a graph, if we do node classification or link prediction, and we want to make new nodes or links, our method must *learn* the graph-based dependencies. Heuristics on the graph could not work very well.

P10: In 2021, we developed data augmentation techniques for graph neural nets. First, look at this node v. We can slightly add or remove edges on v to have a "new" node v'. We find that link prediction models can suggest the addition and removal of edges. We find that training graph neural nets on these new nodes can improve the accuracy of node classification by 17%.

[click] Second, we want to make new labeled node pairs to train graph neural nets for link prediction. So suppose we have an arbitrary node pair u and v. If we can find a confident label, we will use it to augment the training data. So what can be the confident label? Our idea is to find the most similar node pair in the training set, like u' and v'. If the similarity is high enough, we will use the label y' for the node pair u and v. Training on the new node pairs can improve Hits@20 by 16%.

[click] With these two works, we conclude that learning-to-augment methods are effective for graph data. We compared against heuristics-based augmentation. You can find results from the two papers at the bottom of this slide.

P11: OK. Now let's think about the graph regression task for polymers. Can we design learning-to-augment methods for the graph-level tasks, beyond node-level and link-level? Do you remember the three machine learning ideas in polymer data? Yes, supervised learning, imbalanced learning, and transfer learning.

In this NSF project, I work with my PhD students and undergraduates to answer three questions. The first question is graph data augmentation for supervised learning. Given us only 500 labeled graphs, how can we create synthetic data with confident labels?

P12: We propose to use rationalization to make synthetic data. Rationale and environment, these two concepts were used in NLP. Let's look at these two examples for sentiment analysis. In data i, "I've just got home and one of my burgers was stone cold". Here "stone cold" was the rationale. It supports and explains the negative label. The other words are called environment. In data j, this part "has a nice texture" was the rationale of the positive label.

[click] OK, here is my question. Can you make a new example that has a negative label? Any ideas?

[click] Yeah, we can combine i's rationale and j's environment, like this, "I find that my French toast was stone cold." It is a reasonable, negative review.

P13: So let's apply this idea for polymer graphs. We can develop a separation function f_sep. It tells you which nodes are in a rationale subgraph and which are in an environment subgraph. In this example, if we combined i's rationale and j's environment, so we replaced the i's environment by the j's environment, what should be the label? Yeah, the label is y_i. OK? So far it seems good, but!

[click] Combining two chemical subgraphs is scientifically challenging. We cannot guarantee these two graphs can be combined. So what can we do?

P14: We propose to do graph augmentation not in the data space but in latent space. Let's look at our framework. We start from a GNN and MLP that tells you how likely a node is in rationale or environment. m is the vector. The value is between 0

and 1. If m_v is 1.0, v is in the rationale. If it's 0, v is in the environment. OK. Then we can aggregate the node representations to find the vector of rationale (in blue) and the vector of environment (in red).

[click] Now we can do augmentation. We can aggregate the vector of i's rationale h_r_i and the vector of j's environment h_e_j to be a new vector h_i_j. When we train the predictor, we assume this new vector's label is i's label y_i, because the rationale was from graph g_i.

[click] With this framework, we improved the accuracy of molecule classification from .738 to .779 on HIV dataset and from .766 to .819 on BASE which is a drug dataset. We also reduced the error of polymer property prediction from 60.6 to 42.6 on melting temperature and from 770 to 524 on oxygen permeability. We reported results on many other benchmark datasets in the paper. In this work, we find that graph data augmentation can be performed in latent space.

P15: OK, the second thing is imbalanced learning. Basically, we want accurate predictions in minor label areas. So it is important to balance the training label distribution.

P16: Here we can use this 1.2 million unlabeled graphs, then a common idea is self-training. We first train a predictor on the labeled data, this blue one. Then we give unlabeled graphs labels, then we sample confident labels from this orange part. And we merge the blue and orange to make the distribution more balanced.

[click] But, we find that the merged distribution is still quite imbalanced. Why? Two reasons. First, there is naturally fewer data in the minor area. Second, the predictor has low confidence in this area. So how can we make the training set balanced?

[click] Think about this. Let's reverse this distribution and then sample a value. If we can create a graph, whose label is likely this value, we can use this graph to augment the training data. And then the distribution will be balanced, with this green part. OK. This sounds like a good idea, but!

[click] Label-anchored graph decoding is very difficult. We found that given a specific label value, graph generative models were not good enough to decode a nice polymer graph, especially in the minor area. So what can we do?

P17: Again, our idea is to do augmentation in latent space. We propose label-anchored mix-up. First, we split the label space into intervals. Then for each interval, we average the vectors of its labeled graphs to be the interval's vector, like this z_i vector for the i-th interval. We average the labels of its graphs to be the interval's label, like this a_i.

When we have a target label value y, we sample a labeled graph (G_j, y_j). Then we use graph neural nets to get its vector h_j. We also sample an interval and get its label value a_i and vector z_i. Then look at this equation. We calculate this \lambda so the mix of interval label a_i and graph label y_j is exactly the target y. Then we use \lambda to mix the interval vector z_i and the graph vector h_j to get vector h. This h is the vector of our label-anchored graph for label value y. We use (h, y) to augment training data.

P18: This algorithm looks complex but the idea is very simple. If the distribution lacks a label value, we create a vector instead of a graph for this label. Then we have this H_aug, not G_aug. We use the (vector, label) pairs to balance the training data.

OK. We did experiments. We can see our method reduced mean average error on the Free Solvation Database from 1.154 to .777 in the few-shot area. Here the few-shot area is actually the most interesting area. It means the label regions that we have very few training examples. OK. We also reduced the error from .726 to .563 in all areas. We also tested on the polymer melting temperature dataset, and the errors were reduced too.

OK. In this work, we talk about graph imbalanced regression, and we show it again that graph data augmentation can be done in the latent space.

P19: OK. When we show the results to our collaborators, they ask us an interesting question: There are 1.2 million unlabeled polymers. Can we suggest the top three that are most likely "novel polymers"? That means, their labels are above the upper bound. Our collaborators can measure the polymer properties in a wet lab, but they can only do two or three because it's very expensive.

So we did what they said, and we got exciting results. We chose to do thermal conductivity because it's a bit cheaper to measure than other properties. We suggested three polymers. Our collaborators did experiments. They confirmed that two of them were novel. We wrote a paper together about this discovery. And the paper is under review.

Then we propose and get a project funded in the NSF CBET program. Besides the material discovery research, we also propose efforts to organize a machine learning challenge and build a cyber platform for polymer informatics. Our goal is to let the machine learning and polymer science communities know better about each other.

We just proposed a competition to NeurIPS. The competition will focus on two tasks, polymer property prediction and polymer inverse design. Our cyber platform also focuses on these two tasks. To do inverse design, the machine learning models should generate polymer graphs. That's exactly the bottleneck in our past work. We had to compromise and did data augmentation in latent space. Now can we develop a strong generative model for polymer graphs?

P20: Let me introduce our recent breakthrough. We read the DiT paper in 2022 and developed our DiT for graph generation. So DiT is for Scalable Diffusion Models with Transformers. It is one of the important technologies behind Sora. And Sora is the OpenAI product that generates 1 minute video automatically with a text. OK. Last year, before Sora came out, we developed a graph diffusion transformer to generate a new polymer graph with a training example.

You can think like this. Sora is from text to video. Our work is from a polymer property value and a graph to a new polymer. You can find the details in this NeurIPS paper.

[click] In this work, we actually find a new opportunity to learn from unlabeled graphs.

P21: So, what do I mean? OK. Think about this. How do we transfer knowledge from many many unlabeled graphs to downstream? Yes, self-supervised learning. We design reconstruction or perturbation tasks to pre-train the model. I called it parameter-centric transfer. So what's the problem here?

[click] First, the downstream tasks may not appreciate the patterns learned from reconstruction or perturbation. For example, a slight perturbation actually may not hold the property; it can make a toxic thing in-toxic. So pre-training on perturbations can have a negative impact on some downstream tasks. Second, parameter-based transfer is not interpretable. We cannot observe how the knowledge in unlabeled graphs is transferred downstream.

[click] OK. Now we have the graph diffusion transformer. It was first trained on many unlabeled graphs. Then, given a downstream example, it was trained to generate a new graph that has a similar label value. Then we use it to expand training data for downstream prediction.

I want to say, with this work, finally we are able to do graph data augmentation in the data space. The knowledge in unlabeled graphs was extracted by diffusion model, guided by downstream, projected into the generated graphs, and used as training data. So we name our approach Data-Centric Transfer.

P22: Here are some experimental results on molecule classification. These gray cells show that self-supervised learning could make a negative impact. Look, they are smaller than the first line. Many numbers are lower than the accuracy without self-supervised learning.

If you look at the last line, you can see our data-centric transfer achieves the highest accuracy on all the downstream tasks. Because it used graph diffusion transformer to do data augmentation.

In this study, we provide the generated polymers to material scientists, and they give us feedback and insights. They can see and evaluate this transfer and even control it in the loop. Self-supervised learning or pre-training cannot support that.

P23: OK. Let me conclude what we've learned from this materai science project. We find that graph data augmentation can improve polymer property prediction. It can be implicit in vector space. It can be explicit, which is graph generation. And graph data augmentation can be designed for supervised learning, imbalanced learning, and transfer learning.

Finally, the diffusion transformer works for both video creation and material science. Some interesting future directions are, generating polymers on multiple properties, using substructures to guide the generation, and studying if the literature and large language model can provide useful knowledge for polymer research.

P24: OK. Let's take a breath. Hooo… In the next 10 minutes, I'll talk about our project on online education.

P25: In this project we work with UIUC and Gallaudet University. Gallaudet University is dedicated to the education of deaf and hard of hearing students. These students have very different feelings about online lecture videos from ours. They have to rely on video transcription to follow the content but because we professors talk too fast, the transcripts are often broken. And

during the pandemic, many classes were pre-recorded, they had to watch the videos, and they could not know if they understood the lecture content.

[click] In this project, we have two objectives. First, we are developing quiz generation tools and question answering tools. With our tools, instructors can have quizzes for the videos with minimal effort. Second, we are developing a student and classroom simulator to evaluate if the tools can help achieve the learning goals. For the first objective, if we want the QG and QA tools to be useful, the language models behind them must learn knowledge from class material and have good reasoning abilities. At this point, actually we are facing one of the essential problems in NLP.

P26: That is reasoning in natural language. Here we study at least four types of reasoning.

[click] The first is comparative reasoning. Many questions in lectures are comparing concepts and methods. Our EMNLP paper discussed how to improve the language model's ability to compare two documents.

[click] The second is counterfactual reasoning. Teachers sometimes ask questions that have counterfactual presuppositions. Because those questions are a bit more complex and they can test if the students truly remember and understand the lecture content.

For example, in a geography class, we can ask, if the height of Mount Everest dropped by 300 meters, what would be the highest mountain in the world? The answer is K2, also known as Godwin Austen. It's in Pakistan. No matter for a student or a model, if it only associated the highest mountain with Mount Everest, it would fail to answer this question.

Our study on this kind of IfQA problem received an outstanding paper award in EMNLP last year.

OK. The third type is commonsense reasoning.

P27: There are many tasks to test a language model's commonsense. The first is multi-choice QA. For example, the question can be "where can I stand on a river to see water falling without getting wet?" The answer is bridge. The second is fact verification. It is to judge if a statement is true or false. The third is, given you some keywords, generate a sentence. For example, we have dog, frisbee, catch, and throw. We can generate "A girl throws a frisbee and her dog catches it." We don't want "a frisbee throws a dog" or "a dog throws a girl."

The last is making an explanation for a wrong statement. For example, if the statement is "hungry for water", we want to generate why.

We can find benchmark datasets for all these tasks. And all the tasks require commonsense knowledge.

P28: So where can we find commonsense knowledge? We have different types of data. We have commonsense corpus, we have commonsense knowledge graphs, and we have commonsense knowledge bases. So we had a kind of crazy idea - we transformed all these types of data into text. Then we have a large-scale commonsense text dataset. Now we can use retrieval augmentation as a unified approach for all these tasks. We retrieve relevant passages to augment the input of classification models or generation models. If it's generation, it's called RAG, retrieval-augmented generation.

OK. This unified approach is simple and effective. If there are different knowledge sources, we don't have to develop different kinds of retrievers, you know, like text retriever, graph retriever, and table retriever. And we don't have to tune the hyperparameters to combine the multiple retrieval results. So this can be a good solution for other applications too.

P29: Experiments show that our knowledge augmentation framework performs the best on all the four tasks, six datasets. The previous best methods usually designed a specific learning module jointly with a language model. For example, this GreaseLM has a graph neural net to learn from commonsense knowledge graph. It integrates the graph representation and language representation. But our RACo can use a larger collection of data. We use knowledge augmentation to solve this problem. RACo is better than GreaseLM on both CSQA and OBQA.

P30: OK. We have advocated this concept called knowledge-augmented NLP for two years. We also call it open-book approaches. You know, having access to books and lecture notes can improve your exam performance.

We started from a survey paper about knowledge augmentation. We offered three tutorials. And since people learn and use knowledge augmentation, we organize workshops so people can share their results. If you are interested, please submit your work to the third workshop. It is in ACL this summer.

P31: OK. Let's look at the fourth type of reasoning - abductive reasoning. It is to infer the best explanation. So sometimes it is to find intermediate events. For example, if we know we went to work leaving windows open, and we found a mess when we were back home, that might have happened? Maybe a thief broke into our house. Maybe there was a very strong wind. Our ACL paper proposed a mixture of experts model, MoE, on knowledge graphs to generate different kinds of explanations.

[click] OK. One day when I was reading a bunch of papers, I realized we scientists do this comparative reasoning and abductive reasoning all the time. We try to compare existing work to find their common limitations. We compare one idea against the other to tell its advantages. That's comparative reasoning.

And we always look for improvement. We look for a method that performs better than existing work. Our proposed idea is actually an explanation to this improvement. That's abductive reasoning.

P32: OK. Then I thought about a specific task. Can we use language models to generate the sentences starting with the word "However"? You know, these sentences are critical in papers. They tell the limitations of existing work and the motivation of proposed work. Writing "However" sentences is not an easy task. To be honest, not all my students are able to write these "however" sentences correctly in their first year.

So recently we are funded to study how to integrate three technologies for this scientific text intelligence problem. The first is knowledge augmentation. The model must learn or have access to domain knowledge. The second and the third are comparative reasoning and abductive reasoning. I'm very excited about this project. I look forward to some interesting results.

P33: OK. Next I'll use just one page to talk about text generation for security problems. My CRII award found that text and graph representations are complementary in many tasks.

[click] Then two projects are developed based on this conclusion. First, software engineers use concept taxonomies to improve bug traceability. Our task is to build the taxonomy from domain documents. We find that existing work extracts concept terms and then puts them into the taxonomy. And the recall was very low.

Our idea is to "write" the taxonomy like a human. We train a model to learn the existing graph and generate concept names word by word on the taxonomy. The model can make a lot of important general concepts even though they appear rarely in the documents.

[click] The second project is for developers and users of network security systems. We use large language models to interpret the prediction of machine learning models like decision trees to the domain experts in English. The users can easily understand the model's decisions, and they do not have to know much about machine learning.

P34: OK. Lastly, I want to talk about mental health.

P35: My CAREER project is to develop specialized intelligent assistance with generative AI. Specifically it focuses on suicide prevention. On Reddit, Suicidewatch has almost 500 thousand members. People are looking for peer support when they have suicidal thoughts. For example, teenagers may post at midnight about their academic failure, anxiety, and depression. So how can we help them? Can AI suggest to us how to express emotional support effectively? You know AI can suggest how to write a good email, right? Yeah, now we want to see if humans and Generative AI can work together to help humans feel better.

My CAREER also enhances the joint PhD program between ND psychology and computer science. I am supervising a PhD student in this program. At the same time, in an NIH R21 project, I'm collaborating with the psychology professors and clinicians to detect suicide risks in real time. I also learn a lot from this project. If you are interested, we can talk about it offline.

OK. Now I'd like to take 1 minute to talk a bit about my recent thoughts on Generative AI. Everybody is looking forward to OpenAI's GPT-5. We know GPT-4 is already so powerful. So what will be the surprising features of GPT-5?

My prediction is its knowledge and abilities in specialized domains. It can be a master in many areas, from laws to arts, from medicine to surgeries, from biology to psychology, from physics, chemistry to business and finance. I don't know if my prediction is correct. But at least I am happy working on this problem with my collaborators. I mean, develop specialized generative AI for specialized intelligent assistance.

Page 36: OK. Let me conclude my talk. I talked about graph data augmentation for polymer science. I talked about knowledge augmentation for NLP and reasoning. Finally, I want to say, doing interdisciplinary research with generative AI is lots of fun! OK. Now I welcome any questions. Thank you!