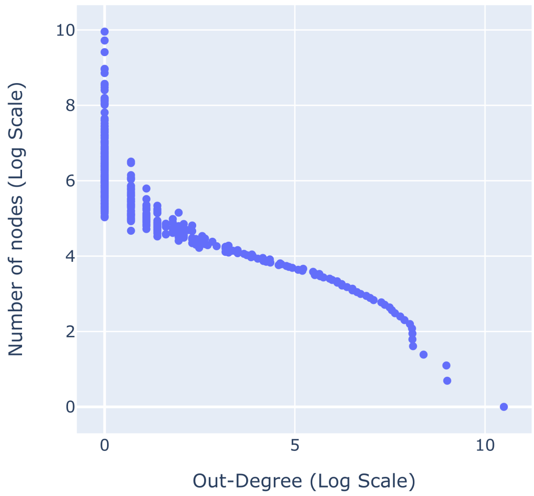

Figure 4: **Out-degree distribution** of all nodes in G_{KG}. Both axis are in log scale.

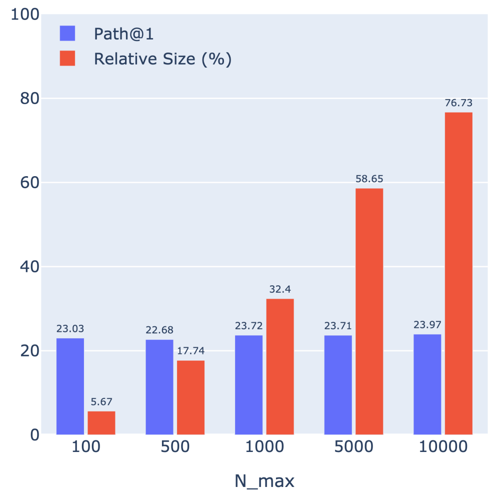

Figure 5: **Effect of subgraph sampling**. Blue bar denotes *path@1* for each N_{max}, while red bar denotes the average relative size of the sampled subgraph compared to the original G_{input}.

A Subgraph Sampling

We explain in more detail the subgraph sampling method adopted by AttnIO, as mentioned in Section 4.

As shown in Figure 4, the out-degree distribution of G_{KG} follows an extreme power-law distribution, which is typical in relational graphs. Among 100K nodes in G_{KG}, about 31K nodes possess only one incoming neighbor node, making the graph extremely sparse. Meanwhile, a node with the highest in-degree has more than 21K incoming neighbor nodes, connecting the node to about 20% of all entities in the whole graph.

We find that the small number of *hub nodes* with high in/out degrees are the major factor that increases the size of input graph G_{input}. Therefore, we choose to limit the maximal number of neighbors to sample from each entity, while constructing G_{input} in the training time. We denote this limit as N_{max}.

The effect of subgraph sampling with different N_{max} is shown in Figure 5. Setting N_{max} to 100, subgraph sampling effectively reduces down the number of edges in the input graph to only 5.67% of the original G_{input} on average, while losing only about 1.0 absolute performance in *path@1*. In all our experiments, we set $N_{max} = 1000$, leaving only 32.4% of the edges originally in G_{input}, while not compromising for the retrieval accuracy.

B Dataset Statistics

Dialog		KG	
# of dialogues:	15,673	$\|V\|$:	100,813
# of turns:	91,209	$\|E\|$:	1,190,658

Table 5: **Dataset Statistics** of OpenDialKG.

The statistics of OpenDialKG dataset is as shown in Table 5. There are 1358 distinct types of relations comprising 1M+ edges.

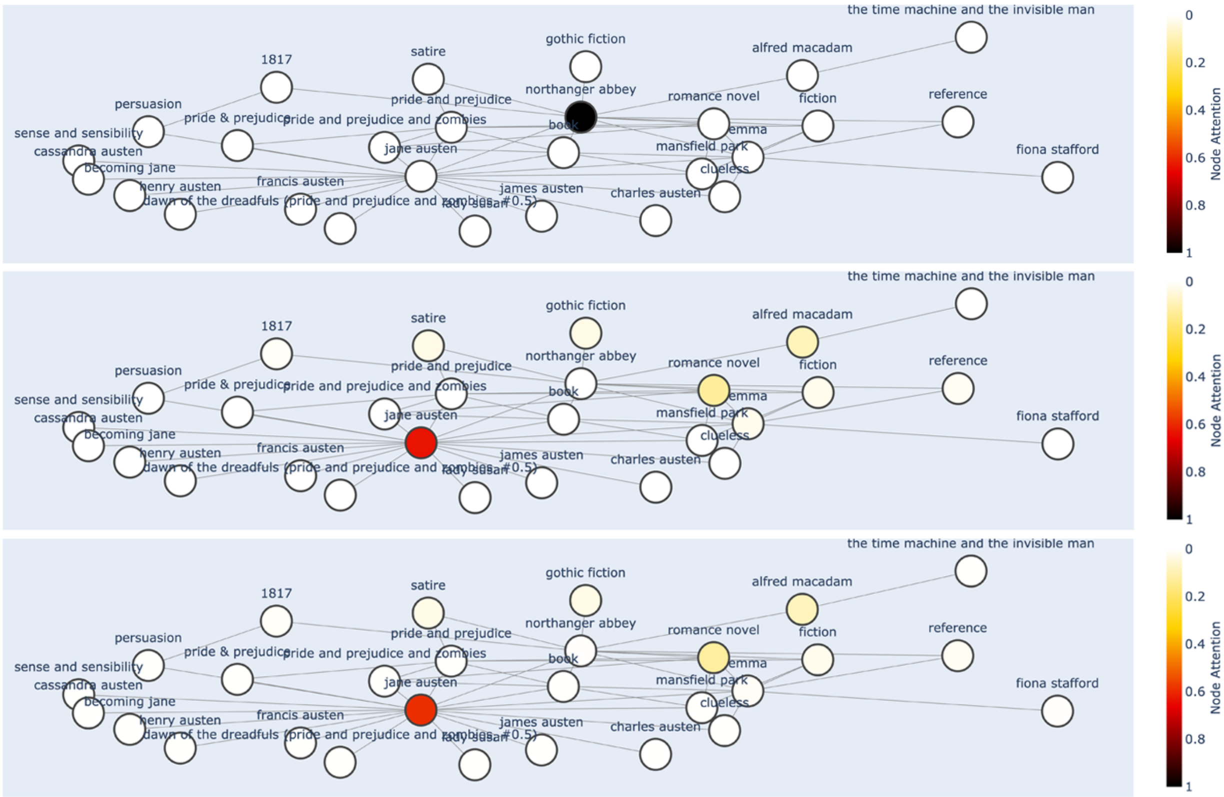

Figure 6: **Node attention visualization** for the additional case study. Each figure represents the node attention at the initial state (Top), after the first decoding step (Center), and after the second decoding step (Bottom). The center and bottom figure might look similar, due to the slight difference of node attention distribution between the two steps. The attention is focused on small number of entities, on the contrary with the even distribution at the former case study.

C Additional Case Study

We provide an additional visualization of node attention distribution for a given dialogue in Figure 6. This time, the model is given a dialogue context as following:

> A: *Someone suggested the Northanger Abbey book to me. Do you know who the author is?*
> B: *[RESPONSE]*

The dialogue is of a similar topic with the case study provided in Section 4.2, but the query in this case is a closed question that specifically asks for the author of a book.

AttnIO first starts from the only initial entity *Northanger Abbey*, and finds from the context that the most relevant relation here is *written by*. Therefore in the first decoding step, AttnIO propagates more than half of the attention value (0.63) from *Northanger Abbey* to its writer, *Jane Austen*. In the second propagation step, AttnIO chooses not to propagate much attention to any of *Jane Austen*'s neighbor nodes, preserving most of the attention value (0.59) by traversing a self-loop. (We omitted self-loops in the visualization for clarity.)

Note that there are a variety of neighbor nodes walkable from *Jane Austen*, just as in the former case study. However, AttnIO understands the intent of user's utterance requiring for a specific answer, concentrating most of the attention to the nodes and edges directly related to the dialogue context.

D Generation Examples

In Table 6, we present more path generation examples along with ground-truth paths for the given dialogues. Note that we only sampled cases where the paths generated from our model are different from the ground-truth paths. Dialogs are partially shown to meet the spatial constraints.

Dialog	A: *I'm not sure who else was in it, but Ralph Fiennes also starred in Wrath of the Titans.* B: *Wrath of the Titans, I didn't know Ralph Fiennes was in that movie. Tell me more about that movie and the stars in it.* A: *[RESPONSE]*
MODEL-AS	*Wrath of the Titans ⇒ starred ⇒ Liam Neeson*
MODEL-TS	*Wrath of the Titans ⇒ starred ⇒ Liam Neeson*
AttnFlow	*Ralph Fiennes ⇒ starred ⇒ The hurt Locker*
GT	*Ralph Fiennes ⇒ starred ⇒ Wrath of the Titans ⇒ written by ⇒ Greg Berlanti*

Dialog	A: *I think Tiger Woods is a good golf player, but is he retired right now?* B: *No he is actually still playing. Is he half asian?* A: *[RESPONSE]*
MODEL-AS	*Asian ⇒ ethnicity of ⇒ Tiger Woods*
MODEL-TS	*Asian ⇒ ethnicity of ⇒ Tiger Woods*
AttnFlow	*Asian ⇒ language ⇒ Vietnamese Language*
GT	*Asian ⇒ includes ⇒ Vietnamese American*

Dialog	A: *Could you recommend books written by Aldous Huxley?* B: *[RESPONSE]*
MODEL-AS	*Aldous Huxley ⇒ wrote ⇒ The doors of perception & heaven and hell*
MODEL-TS	*Aldous Huxley ⇒ wrote ⇒ Brave new world*
AttnFlow	*Aldous Huxley ⇒ cause of death ⇒ Laryngeal Cancer*
GT	*Aldous Huxley ⇒ wrote ⇒ Island*

Dialog	A: *Drew Brees is a quarterback for the new orleans saints. I don't follow football but I hear he is pretty good.* B: *I like movies more than football. I actually liked the american football movies.* A: *[RESPONSE]*
MODEL-AS	*American Football ⇒ subject of ⇒ Wild Cats ⇒ starred actor ⇒ Goldie Hawn*
MODEL-TS	*American Football ⇒ subject of ⇒ Wild Cats ⇒ has genre ⇒ Football*
AttnFlow	*American Football ⇒ sports played ⇒ Troy Aikman*
GT	*American Football ⇒ subject of ⇒ Rudy ⇒ has genre ⇒ Football*

Table 6: **Generated path examples**, along with the ground-truth paths.

E Additional Implementation Detail

Computing Infrastructure	Tesla V100 GPU
Search Strategy	Manual Tuning
Best Validation *path@1*	23.72 (AS), 12.18 (TS)
Training Time (per epoch)	$\approx$ 64min

Hyperparameter	**Search Bound**	**Best Setting**
max path length T	2	2
subgraph sampling limit N_{max}	*choice*[100, 500, 1000, 5000, 10000]	1000
max dialog history	*choice*[3, 4, 5, 6]	3
entity feature dimension	*choice*[60, 80, 100, 120]	80
number of attention heads	*choice*[3, 4, 5, 6]	5
number of epochs	20	20
batch size	*choice*[4, 8, 16]	8
optimizer	*Adam*	*Adam*
learning rate	*loguniform-float*[5e-2, 5e-5]	5e-4
lr scheduler	*reduce_on_plateau*	*reduce_on_plateau*
lr reduction factor	0.1	0.1
gradient clip norm	*uniform-integer[3, 10]*	5

Table 7: **Additional implementation detail** of AttnIO. We follow the specification from Dodge et al. (2019) by reporting hyperparameter search spaces and experimental details.