Disentangling and Integrating Relational and Sensory Information in Transformer Architectures

Awni Altabaa
Department of Statistics & Data Science
Yale University
awni.altabaa@yale.edu

John Lafferty
Department of Statistics & Data Science
Yale University
john.lafferty@yale.edu

Abstract

The Transformer architecture processes sequences by implementing a form of neural message-passing that consists of iterative information retrieval (attention), followed by local processing (position-wise MLP). Two types of information are essential under this general computational paradigm: "sensory" information about individual objects, and "relational" information describing the relationships between objects. Standard attention naturally encodes the former, but does not explicitly encode the latter. In this paper, we present an extension of Transformers where multi-head attention is augmented with two distinct types of attention heads, each routing information of a different type. The first type is the standard attention mechanism of Transformers, which captures object-level features, while the second type is a novel attention mechanism we propose to explicitly capture relational information. The two types of attention heads each possess different inductive biases, giving the resulting architecture greater efficiency and versatility. The promise of this approach is demonstrated empirically across a range of tasks.

1 Introduction

A broad goal of machine learning research is to develop unified architectures capable of learning and reasoning over a wide range of tasks and data modalities, such as text, audio, time-series, and images. The generality of the input and output formats for sequence models such as Transformers (Vaswani et al., 2017) makes them promising candidates for this goal. However, a tension exists between the goal of having a general architecture and the ability to support inductive biases that may be favorable for certain types of tasks. Recent research has shown that the standard Transformer architecture lacks the ability to efficiently learn and represent relational information, leading to several proposals for alternative architectures to encode inductive biases for relational learning (Santoro et al., 2017, 2018; Shanahan et al., 2020; Webb et al., 2021, 2024; Kerg et al., 2022; Altabaa et al., 2024; Altabaa and Lafferty, 2024b). These relational architectures, on the other hand, fail to provide the generality required to handle more general learning tasks. In this paper, we present an extension of the Transformer architecture that preserves the generality of the architecture while integrating inductive biases for processing relational information between objects.

The Transformer can be understood as an instantiation of a broader computational paradigm implementing a neural message-passing algorithm that consists of iterative information retrieval followed by local processing. To process a sequence of objects $x_1, \ldots, x_n$, this takes the general form

$$
\begin{aligned}
x_i &\leftarrow \text{Aggregate}\big(x_i, \{m_{j \to i}\}_{j=1}^n\big) \\
x_i &\leftarrow \text{Process}(x_i)
\end{aligned}
\tag{1}
$$

In the case of Transformers, the self-attention mechanism can be seen as sending messages from object j to object i that are encodings of the sender's features, with the message from sender j to receiver i given by $m_{j \to i} = \phi_v(x_j)$. These messages are then aggregated according to some selection criterion based on the receiver's features, typically given by the softmax attention scores. In
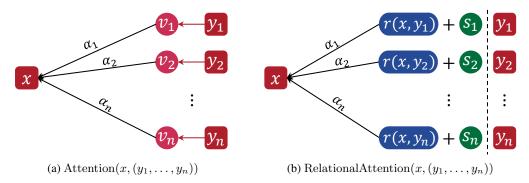
(a) $\text{Attention}(x, (y_1, \ldots, y_n))$ (b) $\text{RelationalAttention}(x, (y_1, \ldots, y_n))$

Figure 1: Standard self-attention retrieves sensory information v_i about the attributes of individual objects while *relational attention* retrieves relational information $r(x, y_i)$ about the relationship between the objects in the context and the receiver. Each relation is tagged with a symbol s_i which acts as an abstract variable identifying the sender. In both cases, information is aggregated according to the attention scores α_i, which are computed by a softmax over inner products of queries and keys.

this work, we argue that there are two distinct types of information that need to be encoded in the messages $m_{j \to i}$. The first we refer to as *sensory information*, which represents attributes or features of individual objects. The second is *relational information* about the relationship between the sender and receiver along various dimensions. The Transformer architecture captures the transmission of sensory information, but does not explicitly support the transmission of relational information.

In this paper, we propose a novel type of attention mechanism that explicitly encodes learned relations between the sender and receiver. For this attention mechanism, the message from the sender object to the receiver object is a set of relations between them, which can be expressed as $m_{j \to i} = r(x_i, x_j)$, with the relations $r(\cdot, \cdot)$ computed through inner products of feature maps. By combining this with the standard attention mechanism of Transformers, we obtain a model that explicitly processes both sensory and relational information by stacking the two types of attention heads with $m_{j \to i} = (\phi_v(x_j), r(x_i, x_j))$. This *Dual Attention* architecture disentangles the two types of information in the aggregation phase of attention, while making both types of information available during the information processing stage.

The contributions in this paper are summarized as follows:

1. We introduce a new *relational attention* mechanism that disentangles sensory information from relational information. While standard self-attention models retrieval of sensory information, relational attention models retrieval of relational information.

2. We introduce a generalized Transformer architecture that integrates sensory and relational information through *Dual Attention*—a form of multi-head attention with two distinct types of attention heads. Standard self-attention heads encode sensory information while relational attention heads encode relational information.

3. We carry out an extensive set of experiments, showing that the *Dual Attention Transformer* outperforms standard Transformers in terms of data efficiency. Our experiments range across several tasks and data modalities, including visual relational reasoning, symbolic mathematical reasoning, language modeling, and image recognition.

2   Disentangling Attention over Sensory and Relational Information

2.1   Standard Attention: Attention over Sensory Information

The attention mechanism of standard Transformers can be understood as a form of neural message-passing that performs selective information retrieval. An object emits a query that is compared against the keys of each object in its context via an inner product. A "match" occurs when the inner product is large, causing the features of the attended object to be retrieved and added to the residual stream of the receiver. Formally, standard attention takes the form

$$\text{Attention}(x, (y_1, \ldots, y_n)) = \sum_{i=1}^{n} \alpha_i(x, \boldsymbol{y}) \phi_v(y_i)$$

$$\alpha(x, \boldsymbol{y}) = \text{Softmax}\left(\left[\langle \phi_q(x), \phi_k(y_i) \rangle \right]_{i=1}^{n} \right),$$

(2)

2

where ϕ_q, ϕ_k are learnable query/key maps controlling the selection criterion and ϕ_v is a learnable value mapping controlling what information is sent. The attention scores $\alpha(x, \boldsymbol{y}) \in \Delta^n$ are used to retrieve a convex combination of the values.

The information being retrieved here is "sensory" information—that is, the features and attributes of individual objects. There is no explicit retrieval of information about the *relationship* between the features of the sender and the receiver. The attention scores $\alpha_i(x, \boldsymbol{y})$ can perhaps be thought of as (normalized) relations between objects, but these are merely computed as an intermediate step in an information-retrieval operation, and are ultimately entangled with the object-level features of the sender[1]. This makes learning relational representations in standard Transformers inefficient.

2.2 Relational Attention: Attention over Relational Information

We propose an attention mechanism with a relational inductive bias. Under the message-passing view of Equation (1), this attention mechanism represents an operation where the message from one object to another encodes the relation between the sender and the receiver. We call this *relational attention*.

At a high level, this operation begins in the same way as the standard attention mechanism, with each object emitting a query and a key, which are compared via an inner product. When the inner product is high, an object is selected. But, now, rather than retrieving the features of the selected object, what is retrieved is the *relation* between the two objects. In addition, we must send an identifier that signals to the receiver "who the sender is". Mathematically, this operation is defined as follows.

$$
\begin{aligned}
\text{RelationalAttention}(x, (y_1, \ldots, y_n)) &= \sum_{i=1}^{n} \alpha_i(x, \boldsymbol{y}) \big(r(x, y_i) W_r + s_i W_s \big), \\
\alpha(x, \boldsymbol{y}) &= \text{Softmax}\big(\left[\langle \phi_q^{\text{attn}}(x), \phi_k^{\text{attn}}(y_i) \rangle \right]_{i=1}^{n} \big) \in \Delta^n, \quad (3) \\
r(x, y) &= \big(\langle \phi_{q,\ell}^{\text{rel}}(x), \phi_{k,\ell}^{\text{rel}}(y) \rangle \big)_{\ell \in [d_r]} \in \mathbb{R}^{d_r}, \\
(s_1, \ldots, s_n) &= \text{SymbolRetriever}(\boldsymbol{y}; \, S_{\text{lib}})
\end{aligned}
$$

Thus, relational attention between the object x and the context $\boldsymbol{y} = (y_1, \ldots, y_n)$ retrieves a convex combination of x's relations with each object in the context, $r(x, y_i)$, each tagged with a symbol s_i that identifies the sender. We will expand on the role and implementation of the symbols in the next subsection. Here, $\phi_q^{\text{attn}}, \phi_k^{\text{attn}}$ are learned feature maps that control the selection criterion for which object(s) in the context to attend to. Another set of query/key feature maps, $\phi_{q,\ell}^{\text{rel}}, \phi_{k,\ell}^{\text{rel}}, \ell \in [d_r]$, are learned to represent the relation between the sender and the receiver. Each inner product $\langle \phi_{q,\ell}^{\text{rel}}(x), \phi_{k,\ell}^{\text{rel}}(y) \rangle$ can be thought of as a comparison of the two objects' features under a particular filter—i.e., a 'relation'. The d_r pairs of feature maps produce a d_r-dimensional relation vector.

In some tasks, a good inductive bias on the relations is *symmetry*: the relation between x and y is the same as the relation between y and x. This can be achieved by using the same feature filter for the query and key maps (i.e., $\phi_q^{\text{rel}} = \phi_k^{\text{rel}}$). This imbues the relations with added structure, making them positive semi-definite kernels which define a pseudometric on the object space and a corresponding geometry. We explore this and expand on this discussion in our experiments.

2.3 Symbol Assignment Mechanisms

For the purposes of processing relations, the receiver needs to know: 1) the relation between itself and the objects in its context, and 2) the identity of the object corresponding to each relation. The symbols in relational attention are used to tag each relation with the identity of the sender (the object the relation is with). Without this information, the result of relational attention would only be an aggregated representation of the relations between the receiver and the selected object(s).

A symbol identifies or "points to" an object, but, importantly, it does not fully encode the features of the object. The second point is what makes relational attention (Equation (3)) "disentangled" from sensory or object-level features. The sensory features of individual objects are high-dimensional and have a lot of variability. By contrast, relational information is low-dimensional and more abstract. If

---

[1]In principle, it is also possible that the MLP compute a relation between the sender and receiver in the local processing step by separating their representations and computing a comparison. However, this is difficult to do since the representations of the objects will be additively mixed, and there is no inductive bias pressuring the computed function to be a relation.

sensory features are mixed with relational information, the sensory information could overwhelm the relational information, preventing abstraction and generalization. Instead, symbols act as abstract references to objects and may be thought of as connectionist analogs of pointers in traditional symbolic architectures.

Here, the "identity" of an object may mean different things in different situations. For us, identity may be encoded by 1) position, 2) relative position, or 3) an equivalence class over features. Correspondingly, we consider three different symbol assignment mechanisms.

A symbol identifies or "points to" an object, but, importantly, does not fully encode the features of the object. The second point is what makes relational attention (Equation (3)) "disentangled" from sensory or object-level features. The sensory features of individual objects are high-dimensional and have a lot of variability. By contrast, relational information is low-dimensional and more abstract. If sensory features are mixed with relational information, they would overwhelm the relational information, preventing abstraction and generalization. Instead, symbols act as abstract references to objects, perhaps thought of as a connectionist analog of pointers in traditional symbolic architectures.

Here, the "identity" of an object may mean different things in different situations. For us, identity may be encoded by 1) position, 2) relative position, or 3) an equivalence class over features. We consider three different corresponding symbol assignment mechanisms.

Positional Symbols. In some applications, it is sufficient to identify objects through their position in the input sequence. We maintain a library of symbols $S_{\text{lib}} = (s_1, \ldots, s_{\texttt{max\_len}}) \in \mathbb{R}^{\texttt{max\_len} \times d_{\text{model}}}$ and assign s_i to the i-th object in the sequence. These are essentially learned positional embeddings.

Position-Relative Symbols. Sometimes, the more useful identifier is *relative* position with respect to the receiver, rather than absolute position. This can be implemented with position-relative embeddings. We learn a symbol library $S_{\text{lib}} = (s_{-\Delta}, \ldots, s_{-1}, s_0, s_1, \ldots, s_\Delta) \in \mathbb{R}^{(2\Delta+1) \times d_{\text{model}}}$, and relational attention becomes $x_i' \leftarrow \sum_j \alpha_{ij}(r(x_i, x_j)\, W_r + s_{j-i}\, W_s)$, with "$m_{j \to i} = (r(x_i, x_j), s_{j-i})$.

Symbolic Attention. In certain domains, some information about the objects' features is necessary for identifying them for the purposes of relational processing. Yet, to maintain a relational inductive bias, we would like to avoid sending a full encoding of object-level features. In symbolic attention, we learn a set of symbol vectors, $S_{\text{lib}} = (s_1, \ldots, s_{n_s}) \in \mathbb{R}^{n_s \times d_{\text{model}}}$ and a matching set of feature templates $F_{\text{lib}} = (f_1, \ldots, f_{n_s})$. We retrieve a symbol for each object by an attention operation that matches the input vectors x_i against the feature templates f_j and retrieves symbols s_j.

$$\text{SymbolicAttention}(\boldsymbol{x}) = \text{Softmax}\big((\boldsymbol{x}\, W_q)\, F_{\text{lib}}^\top\big) S_{\text{lib}}. \tag{4}$$

Here, $S_{\text{lib}}, F_{\text{lib}}, W_q$ are learned parameters. This can be thought of as implementing a learned differentiable "equivalence class map" over feature embeddings. Crucially, the number of symbols (i.e., "feature equivalence classes") is *finite*, which enables relational attention to still produce a relation-centric representation while tagging the relations with the necessary identifier.

We find that different symbol assignment mechanisms are more effective in different domains.

2.4   Approximation theory: What Class of Functions can Relational Attention Compute?

To get some intuition about the type of computation that relational attention can perform, we present the following approximation result. The following theorem says that relational attention can approximate any function on $\mathcal{X} \times \mathcal{Y}^n$ which 1) selects an element in $(y_1, \ldots, y_n)$, then 2) computes a relation with it. Both the selection criterion and the relation function are arbitrary, and the selection criterion is query-dependent. The formal statement and proof are given in Appendix A.

Theorem 1 (Informal). *Let* $\text{Select} : \mathcal{X} \times \mathcal{Y}^n \to \mathcal{Y}$ *be an arbitrary preference selection function, which selects an element among* $(y_1, \ldots, y_n)$ *based on a query-dependent preorder relation* $\{\preccurlyeq_x\}_{x \in \mathcal{X}}$. *Let* $\text{Rel} : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}^{d_r}$ *be an arbitrary continuous relation function on* $\mathcal{X} \times \mathcal{Y}$. *There exists a relational attention module that approximates the function* $\text{Rel}(x, \text{Select}(x, \boldsymbol{y}))$ *to arbitrary precision.*

3   Integrating Attention over Sensory and Relational Information

3.1   Dual Attention

One of the keys to the success of the Transformer architecture is the use of so-called *multi-head* attention. This involves learning and computing multiple attention operations in parallel at each layer

4

and concatenating the output. This enables learning multiple useful criteria for routing information between objects. However, in standard Transformers, all these attention heads share the same inductive bias, focusing on sensory information about individual objects. In particular, there is no explicit support for processing *relational* information between objects.

We posit that sensory and relational information are the two primary types of information that are of relevance when processing sequences or collections of objects. In this paper, we explore the effects of augmenting a Transformer with a specialized attention operation with relational inductive biases. We propose a type of multi-head attention with two distinct types of attention heads: standard self-attention, and relational attention. Our hypothesis is that by having both kinds of computations available to the model, it can learn to use both and select between them depending on the current task or context.

Algorithm 1 describes the proposed module. The number of self-attention heads n_h^{sa} and number of relational attention heads n_h^{ra} are hyperparameters. The self-attention heads attend to and retrieve sensory information while the relational attention heads attend to and retrieve relational information. The $n_h = n_h^{sa} + n_h^{ra}$ heads are then concatenated to produce the output. The result is a representation of contextual information with integrated sensory and relational components. We note that a symmetry inductive bias can be injected into the relations $\boldsymbol{r}_{ij}$ by imposing that $W_q^{\mathrm{rel}} = W_k^{\mathrm{rel}}$.

Algorithm 1: Dual Attention

Input: $\boldsymbol{x} = (x_1, \ldots, x_n) \in \mathbb{R}^{n \times d}$

`Compute self-attention heads`

$$\boldsymbol{\alpha}^{(h)} \leftarrow \mathrm{Softmax}\big((\boldsymbol{x}\,W_q^h)(\boldsymbol{x}\,W_k^h)^{\mathsf{T}}\big), \qquad h \in [n_h^{sa}]$$

$$e_i^{(h)} \leftarrow \sum_j \alpha_{ij}^{(h)} x_j\, W_v^h, \qquad i \in [n], h \in [n_h^{sa}]$$

$$e_i \leftarrow \mathrm{concat}\big(e_i^{(1)}, \ldots, e_i^{(n_h^{sa})}\big)\, W_o^{sa}, \qquad i \in [n]$$

`Assign symbols:` $\boldsymbol{s} = (s_1, \ldots, s_n) \leftarrow \mathrm{SymbolRetriever}(\boldsymbol{x}; S_{\mathrm{lib}})$
`Compute relational attention heads`

$$\boldsymbol{\alpha}^{(h)} \leftarrow \mathrm{Softmax}\big((\boldsymbol{x}\,W_{q,h}^{\mathrm{attn}})(\boldsymbol{x}\,W_{k,h}^{\mathrm{attn}})^{\mathsf{T}}\big), \qquad h \in [n_h^{ra}]$$

$$\boldsymbol{r}_{ij} \leftarrow \big(\,\langle x_i\, W_{q,\ell}^{\mathrm{rel}}, x_j\, W_{k,\ell}^{\mathrm{rel}}\rangle\,\big)_{\ell \in [d_r]} \qquad i, j \in [n]$$

$$a_i^{(h)} \leftarrow \sum_j \alpha_{ij}^{(h)}\big(\boldsymbol{r}_{ij}\, W_r^h + s_j\, W_s^h\big), \qquad i \in [n],\ h \in [n_h^{ra}]$$

$$a_i \leftarrow \mathrm{concat}\big(a_i^{(1)}, \ldots, a_i^{(n_h^{ra})}\big) W_o^{ra}, \qquad i \in [n]$$

Output : $\big(\mathrm{concat}(e_i, a_i)\big)_{i=1}^n$

Attention Masks & Causality. Any type of attention mask (e.g., causal mask for autoregressive language modeling) can be implemented in relational attention in the same way as for standard self-attention (i.e., mask is added to α_{ij}^h pre-softmax).

Positional Encoding. There exists different methods in the literature on encoding positional information in the Transformer architecture. For example, Vaswani et al. (2017) propose adding positional embeddings to the input, Shaw et al. (2018) propose adding relative-positional embeddings at each attention operation, and Su et al. (2023) propose rotary positional embeddings (RoPE) which apply a position-dependent map to the queries and keys pre-softmax. These methods are compatible with dual attention and are configurable options in our public implementation.

Computational complexity. The computational complexity of relational attention scales the same as self-attention with a $O(n^2)$ dependence on sequence length. Like standard attention, relational attention can be computed in parallel via efficient matrix multiplication operations.

3.2 The Dual Attention Transformer Architecture

A standard Transformer implements a procedure of iterative information retrieval (attention) followed by local processing (MLP). We define our Dual Attention Transformer in the same way, except that self-attention is replaced by dual attention. Algorithms 2 and 3 defines an encoder and decoder block with dual attention. Composing these blocks yields the Dual Attention Transformer.

Remark 1. *In our implementation, the symbol library S_{lib} is shared across layers to reduce the number of parameters. The rationale is that, since the role of symbols is merely to act as abstract variables, they can be remapped at each layer (e.g., a variable can be reassigned to point to a different object).*

Algorithm 2: Dual Attention Encoder Block

Input : $x \in \mathbb{R}^{n \times d}$
$x \leftarrow \text{Norm}(x + \text{DualAttn}(x))$
$x \leftarrow \text{Norm}(x + \text{MLP}(x))$

Output: x

Algorithm 3: Dual Attention Decoder Block

Input : $x, y \in \mathbb{R}^{n \times d}$
$x \leftarrow \text{Norm}(x + \text{DualAttn}(x))$
$x \leftarrow \text{Norm}(x + \text{CrossAttn}(x, y))$
$x \leftarrow \text{Norm}(x + \text{MLP}(x))$
Output : x

An encoder-decoder architecture with causal dual-head attention in the decoder can be applied to sequence-to-sequence tasks, as in the original Transformer paper (Vaswani et al., 2017). An encoder-only architecture can be used for a BERT-style language embedding model (Devlin et al., 2019) or a Vision Transformer-style vision model (Dosovitskiy et al., 2021). A decoder-only architecture with causal dual-head attention can be used for autoregressive language modeling[2].

We note that a standard Transformer is a special case of this architecture where $n_h^{sa} = n_h, n_h^{ra} = 0$. Appendix B provides further discussion on the details of the architecture and its implementation.

4 Empirical evaluation

We empirically evaluate the Dual Attention Transformer (abbreviated *DAT*) architecture on a range of tasks covering different domains and modalities. For each experiment, we fix the total number of heads, and compare different configurations of *DAT* against a standard Transformer where all heads are self-attention heads. The difference in performance can be interpreted as indicating the effect of having two types of attention heads integrating sensory and relational information. Further experimental details can be found in Appendix C.

4.1 Sample Efficient Relational Reasoning: Relational Games

We begin our empirical evaluation with a benchmark contributed by Shanahan et al. (2020) for evaluating the relational reasoning capabilities of machine learning models. The dataset, called "Relational Games", consists of a family of binary classification tasks, each testing a model's ability to identify a particular visual relationship among a series of objects. The input is an RGB image depicting a grid of objects, and the target is a binary classification indicating whether the particular relation holds for this input. We use this suite of benchmarks to evaluate the *sample efficiency* of *DAT* compared to a standard Transformer. We find that *DAT* is significantly more sample-efficient, particularly at more difficult tasks.

Since the input is an image, we use a Vision Transformer-type architecture (Dosovitskiy et al., 2021) where the input image is split up into patches and then fed into the model as a sequence. We fix the total number of attention heads to 2. We compare a Vision Transformer with $n_h^{sa} = 2$ to two configurations of *DAT*: one with $n_h^{sa} = n_h^{ra} = 1$ and one with $n_h^{sa} = 0, n_h^{ra} = 2$.

We evaluate learning curves by varying the size of the training set, training each model until convergence, and evaluating on a hold-out validation set. We repeat this 5 times with different random seeds to compute approximate confidence intervals. This is depicted in Figure 2. We find that both configurations of *DAT* are consistently more sample-efficient compared to the standard Transformer. The effect is particularly dramatic on the 'match pattern' task which is the most difficult and requires identifying a "second-order" relation (a relation between relations).

---

[2]"Decoder-only" is the commonly used term for this type of architecture (Radford et al., 2018), but it can also be viewed as an encoder-only architecture with causal attention
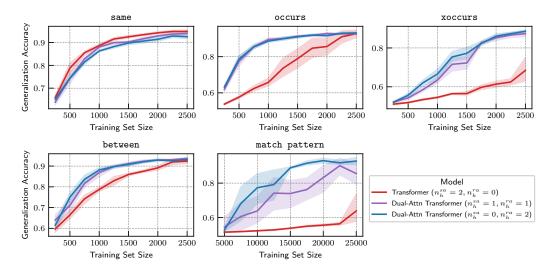
Figure 2: Learning curves on the relational games benchmark. *DAT* is more sample-efficient compared to a Transformer with the same total number of heads. Solid lines indicate the mean over 5 trials with different random seeds and the shaded regions indicate bootstrap 95% confidence intervals.

In this experiment, we use positional symbols as the symbol assignment mechanism since the objects can be identified through their position on the grid. We also impose symmetry on the relations in relational attention, which we find to be a useful inductive bias. Intuitively, this is because the task-relevant relations are symmetric similarity relations across different visual attributes. We provide further discussion and present ablations in Appendix C.1.

4.2 Improved Symbolic Reasoning in Sequence-to-Sequence tasks: Mathematical Problem Solving

Next, we evaluate *DAT* on a set of mathematical problem-solving tasks based on the benchmark contributed by Saxton et al. (2019). We use this as a proxy for "symbolic reasoning". Mathematical problem-solving is an interesting test for neural models because it requires more than statistical pattern recognition—it requires inferring laws, axioms, and symbol manipulation rules. The benchmark consists of a suite of mathematical problem-solving datasets, with each dataset consisting of a set of question-answer pairs. The tasks range across several modules or topics including solving equations, adding polynomials, expanding polynomials, differentiating functions, predicting the next term in a sequence, etc. For example, an example of question in the 'polynomials__expand' task is "Expand (5*x - 3) * (2*x + 1)" with the target "10 * x ** 2 - x - 3".

This is modeled as a sequence-to-sequence task with character-level encoding. We compare *DAT* against a Transformer using matching encoder-decoder architectures. We use 2-layer models with the total number of heads fixed to 8 in both the encoder and the decoder. We compare an encoder-decoder Transformer with $n_h^{sa} = 8$ against two configurations of *DAT*: one with $n_h^{sa} = n_h^{ra} = 4$ for the encoder and $n_h^{sa} = 8, n_h^{ra} = 0$ for the decoder (config 1) and another with $n_h^{sa} = n_h^{ra} = 4$ for both the encoder and decoder (config 2). The number of cross-attention heads is 8 in all cases. The *DAT* models use position-relative symbols as their symbol assignment mechanism.

Each model is trained for 100 epochs, and accuracy on a hold-out validation set is tracked over the course of training. For each model and task, we run 5 trials with different random seeds to compute approximate confidence intervals. We find that *DAT* models learn faster and reach higher accuracies compared to a standard Transformer.

4.3 Improvements in Language Modeling

In this section, we evaluate *DAT* on autoregressive language modeling. Transformer language models are typically built on what is sometimes called a "decoder-only" architecture. The model receives a sequence of tokens as input and is trained to causally predict the next token at each position.
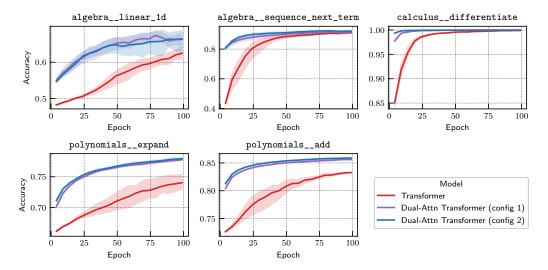
7

Figure 3: Validation accuracy over the course of training on mathematical problem-solving tasks. *DAT* learns faster and reaches higher accuracy. Solid lines indicate mean over 5 trials with different random seeds, and shaded regions indicate 95% bootstrap confidence intervals.

We evaluate the language modeling capabilities of *DAT*, as compared to standard Transformers, using the "Tiny Stories" dataset of Eldan and Li (2023). The dataset consists of short stories and is intended as a benchmark for small language models. Again, for each configuration, we fix the total number of attention heads, and compare a Transformer with only standard self-attention heads to *DAT* models with a mix of self-attention and relational attention heads. We compare a Transformer with $n_h^{sa} = 8$ attention heads to two configurations of *DAT*, one with $n_h^{sa} = 6, n_h^{ra} = 2$ and another with $n_h^{sa} = n_h^{ra} = 4$.
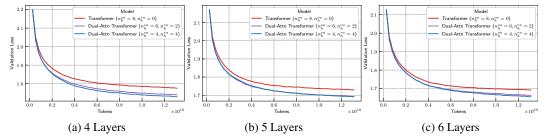

Figure 4: Validation loss curves on a language modeling task. The x-axis indicates the number of tokens and the y-axis is the validation loss. *DAT* achieves a smaller validation loss for the same total number of attention heads.

Figure 4 depicts the validation loss throughout training for each model. We find that *DAT* models with dual head attention achieve lower loss for the same total number of attention heads. We also varied the number of layers, and observed that the trend persists as the number of layers increases. The effect is small but consistent. The two *DAT* configurations behave similarly, with perhaps a very slight advantage to $n_h^{sa} = n_h^{ra} = 4$ (the configuration with a balanced composition of head types).

In Figure 4, the *DAT* models use *symbolic attention* as the symbol assignment mechanism and asymmetric relations in relational attention. We find that symbolic attention outperforms position-relative symbols on this language modeling task. In fact, with position-relative symbols, there is no discernable advantage over the Transformer. Symbolic attention may be well-suited to language due to its implementation of a learned differentiable equivalence class mapping, which can perhaps be thought of as a form of syntax. We also find that asymmetric relations in relational attention perform better than symmetric relations. This may be because the relevant relations in language modeling are asymmetric (e.g., asymmetric syntactic or grammatical relations such as noun-verb, subject-object, determiner-noun, etc.). We provide further discussion and present ablations in Appendix C.3.

We conclude this section by noting that modern large language models are applied to diverse and multi-modal tasks, where different inductive biases will be useful in different contexts. While the language models explored in this section are small, an interesting avenue for future research would be to investigate whether the observed performance benefits scale up to larger models.

4.4 The Benefits of Relational Inductive Biases in Vision: Image Recognition with ImageNet

In the final set of experiments, we evaluate *DAT* on a vision task—object classification with the ImageNet dataset (Russakovsky et al., 2015). This further probes *DAT*' ability in different modalities as a general-purpose sequence model. This section also stress tests *DAT* at larger scales.
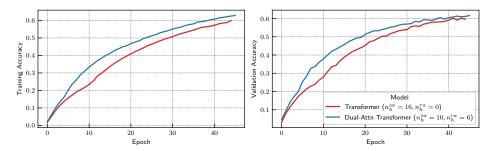

Figure 5: *DAT* compared to a Vision Transformer on image recognition with ImageNet. *DAT* learns faster and achieves better performance.

Here, we again use a Vision Transformer-style architecture (Dosovitskiy et al., 2021). ImageNet's RGB images are divided into 16×16 patches, flattened, and linearly embedded into a vector. A learnable positional embedding is added to each patch embedding. We also prepend a special classification token. The sequence of patch embeddings is then fed through an Encoder and the embedding of the class token is used to generate the final classification through a fully connected layer. We compare a Vision Transformer model with $n_h^{sa} = 16$ to an *DAT* model with $n_h^{sa} = 10, n_h^{sa} = 6$. For both, we used a model dimension $d_{\text{model}} = 1024$ and $L = 24$ layers. The *DAT* model uses position-relative symbols as the symbol assignment mechanism and symmetric relational attention.

Figure 5 depicts the training and validation accuracy over the course of training. We find that *DAT* learns significantly faster. Averaging over epochs, *DAT* has 5.0 (resp., 4.4) percentage points higher training accuracy (resp., validation accuracy) over the course of training compared to a standard Vision Transformer. At the end of training, *DAT* maintains a 2.9 (resp., 1.5) percentage point advantage. This suggests that relational processing is important in processing visual scenes. This matches our intuition that parsing a visual scene requires reasoning about the visual relations between different objects or parts in the scene.

5 Discussion

Summary. The standard attention mechanism of Transformers provides a versatile mechanism for retrieval of sensory information in any given context, but does not explicitly support retrieval of relational information. In this work, we presented an extension of the Transformer architecture that disentangles and integrates sensory and relational information through a variant of multi-head attention with two distinct types of attention heads: standard self-attention for sensory information and a novel *relational attention* mechanism for relational information. We empirically evaluate this architecture and find that it yields performance improvements across a range of tasks and modalities.

Limitations. The proposed architecture introduces several hyperparameters and possible configurations. Although we carried out ablations on the major configuration choices (e.g., composition of head types, symmetry, symbol assignment mechanism), a more thorough empirical examination would help develop an improved understanding of the behavior of this architecture under different configurations. Such a systematic study may also enable the discovery of further modifications to improve the architecture. We also note that our implementation of the Dual-Attention Transformer lacks the hardware-aware optimizations of standard Transformers (Dao et al., 2022), making it slower. However, we expect that similar optimizations are possible for dual attention and *DAT*.

9

Code and Reproducibility

Our code is publicly available at https://github.com/Awni00/abstract_transformer. It includes an implementation of the Dual Attention Transformer, instructions for reproducing our experiments, and links to detailed experimental logs.

Acknowledgment

References

Altabaa, Awni and John Lafferty (2024a). "Approximation of Relation Functions and Attention Mechanisms". arXiv: 2402.08856 [cs, stat] (cited on page 13).

Altabaa, Awni and John Lafferty (2024b). "Learning Hierarchical Relational Representations through Relational Convolutions". arXiv: 2310.03240 [cs] (cited on page 1).

Altabaa, Awni, Taylor Whittington Webb, Jonathan D. Cohen, and John Lafferty (2024). "Abstractors and relational cross-attention: An inductive bias for explicit relational reasoning in Transformers". In: *The Twelfth International Conference on Learning Representations* (cited on pages 1, 20–23).

Ba, Jimmy Lei, Jamie Ryan Kiros, and Geoffrey E. Hinton (2016). "Layer Normalization". arXiv: 1607.06450 [stat.ML] (cited on page 15).

Dao, Tri, Dan Fu, Stefano Ermon, Atri Rudra, and Christopher Ré (2022). "Flashattention: Fast and memory-efficient exact attention with IO-awareness". In: *Advances in Neural Information Processing Systems* (cited on page 9).

Dauphin, Yann N, Angela Fan, Michael Auli, and David Grangier (2017). "Language modeling with gated convolutional networks". In: *International conference on machine learning*. PMLR (cited on page 15).

Debreu, Gerard et al. (1954). "Representation of a preference ordering by a numerical function". In: *Decision processes* (cited on page 12).

Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova (2019). "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding". In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics (cited on page 6).

Dosovitskiy, Alexey et al. (2021). "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale". In: *International Conference on Learning Representations* (cited on pages 6, 9, 20).

Eldan, Ronen and Yuanzhi Li (2023). "TinyStories: How Small Can Language Models Be and Still Speak Coherent English?" arXiv: 2305.07759 [cs] (cited on pages 8, 17).

Hendrycks, Dan and Kevin Gimpel (2016). "Gaussian Error Linear Units (GELUs)". arXiv: 1606.08415 [cs.LG] (cited on page 15).

Inan, Hakan, Khashayar Khosravi, and Richard Socher (2017). "Tying Word Vectors and Word Classifiers: A Loss Framework for Language Modeling". In: *International Conference on Learning Representations* (cited on page 18).

Kerg, Giancarlo, Sarthak Mittal, David Rolnick, Yoshua Bengio, Blake Richards, and Guillaume Lajoie (2022). "On Neural Architecture Inductive Biases for Relational Tasks". arXiv: 2206.05056 [cs] (cited on page 1).

Locatello, Francesco, Dirk Weissenborn, Thomas Unterthiner, Aravindh Mahendran, Georg Heigold, Jakob Uszkoreit, Alexey Dosovitskiy, and Thomas Kipf (2020). "Object-Centric Learning with Slot Attention". arXiv: 2006.15055 [cs, stat] (cited on page 15).

Radford, Alec, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever (2018). "Improving Language Understanding by Generative Pre-Training". In: (cited on page 6).

Russakovsky, Olga et al. (2015). "ImageNet Large Scale Visual Recognition Challenge". In: *International Journal of Computer Vision (IJCV)* (cited on pages 9, 20).

Santoro, Adam, Ryan Faulkner, David Raposo, Jack Rae, Mike Chrzanowski, Theophane Weber, Daan Wierstra, Oriol Vinyals, Razvan Pascanu, and Timothy Lillicrap (2018). "Relational Recurrent Neural Networks". In: *Advances in Neural Information Processing Systems*. Curran Associates, Inc. (cited on page 1).

Santoro, Adam, David Raposo, David G. T. Barrett, Mateusz Malinowski, Razvan Pascanu, Peter Battaglia, and Timothy Lillicrap (2017). "A Simple Neural Network Module for Relational Reasoning". arXiv: 1706.01427 [cs] (cited on page 1).

Saxton, David, Edward Grefenstette, Felix Hill, and Pushmeet Kohli (2019). "Analysing Mathematical Reasoning Abilities of Neural Models". In: *International Conference on Learning Representations* (cited on pages 7, 16, 17).

Shanahan, Murray, Kyriacos Nikiforou, Antonia Creswell, Christos Kaplanis, David Barrett, and Marta Garnelo (2020). "An Explicitly Relational Neural Network Architecture". In: *Proceedings of the 37th International Conference on Machine Learning* (cited on pages 1, 6).

Shaw, Peter, Jakob Uszkoreit, and Ashish Vaswani (2018). "Self-Attention with Relative Position Representations". In: *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*. Ed. by Marilyn Walker, Heng Ji, and Amanda Stent. New Orleans, Louisiana: Association for Computational Linguistics (cited on page 5).

Shazeer, Noam (2020). "GLU Variants Improve Transformer". arXiv: 2002.05202 [cs, stat] (cited on page 15).

Su, Jianlin, Yu Lu, Shengfeng Pan, Ahmed Murtadha, Bo Wen, and Yunfeng Liu (2023). "RoFormer: Enhanced Transformer with Rotary Position Embedding". arXiv: 2104.09864 [cs] (cited on pages 5, 15).

Touvron, Hugo et al. (2023). "Llama 2: Open Foundation and Fine-Tuned Chat Models". arXiv: 2307.09288 [cs] (cited on pages 15, 18).

Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin (2017). "Attention Is All You Need". In: *Advances in neural information processing systems* (cited on pages 1, 5, 6, 15).

Webb, Taylor W., Ishan Sinha, and Jonathan D. Cohen (2021). "Emergent Symbols through Binding in External Memory". arXiv: 2012.14601 [cs] (cited on page 1).

Webb, Taylor W. et al. (2024). "The Relational Bottleneck as an Inductive Bias for Efficient Abstraction". In: *Trends in Cognitive Sciences* (cited on page 1).

Xiong, Ruibin, Yunchang Yang, Di He, Kai Zheng, Shuxin Zheng, Chen Xing, Huishuai Zhang, Yanyan Lan, Liwei Wang, and Tieyan Liu (2020). "On layer normalization in the transformer architecture". In: *International Conference on Machine Learning*. PMLR (cited on page 15).

Zhang, Biao and Rico Sennrich (2019). "Root mean square layer normalization". In: *Advances in Neural Information Processing Systems* (cited on page 15).

A Function Class of Relational Attention: a universal approximation result

To gain a better understanding of the types of functions that can be computed by relational attention, we presented a simple approximation result (Theorem 1) in Section 2.4. Here, we will provide a formal statement of the result and prove it.

Recall that relational attention is a mapping on $\mathbb{R}^d \times \mathbb{R}^{n \times d} \to \mathbb{R}^{d_{\text{out}}}$, where d is the dimension of the input objects and d_{out} is the output dimension. For convenience, we denote the "query space" by $\mathcal{X}$ and the "key space" by $\mathcal{Y}$, though both are $\mathbb{R}^d$ in this setting. Relational attention takes as input a query $x \in \mathcal{X}$ and a collection of objects $\boldsymbol{y} = (y_1, \ldots, y_n) \in \mathcal{Y}^n$ and computes the following

$$\text{RA}(x, \boldsymbol{y}) = \sum_{i=1}^{n} \alpha_i(x; \boldsymbol{y}) \big(r(x, y_i) W_r + s_i W_s \big), \tag{5}$$

$$\alpha(x; \boldsymbol{y}) = \text{Softmax}\Big(\big[\left\langle \phi_q^{\text{attn}}(x), \phi_k^{\text{attn}}(y_i) \right\rangle \big]_{i=1}^{n} \Big) \in \Delta^n, \tag{6}$$

$$r(x, y_i) = \big(\left\langle \phi_{q,\ell}^{\text{rel}}(x), \phi_{k,\ell}^{\text{rel}}(y_i) \right\rangle \big)_{\ell \in [d_r]} \in \mathbb{R}^{d_r}, \tag{7}$$

$$(s_1, \ldots, s_n) = \text{SymbolRetriever}\left(\boldsymbol{y}; S_{\text{lib}}\right) \in \mathbb{R}^{n \times d_{\text{out}}}, \tag{8}$$

where $\phi_q^{\text{attn}}, \phi_k^{\text{attn}}, \phi_{q,\ell}^{\text{rel}}, \phi_{k,\ell}^{\text{rel}} : \mathbb{R}^d \to \mathbb{R}^{d_k}$ are the feature maps defining the attention mechanism and the relation, respectively. For this section, these are multi-layer perceptrons. Note that in Algorithm 1 these are linear maps, but they are preceded by multi-layer perceptron in Algorithms 2 and 3, which makes the overall function class the same. Moreover, for this analysis we will take $W_r = I, d_{\text{out}} = d_r$ and $W_s = 0$. We will later discuss how the role of symbols fits within the message of the result.

The following result states that relational attention can approximate any function of the form: 1) select an object in $(y_1, \ldots, y_n)$ by an arbitrary query-dependent selection criterion, and 2) compute an arbitrary relation $r : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}^{d_r}$ with the selected object. This is formalized below.

To formalize (1), we adopt an abstract and very general formulation of a "selection criterion" in terms of family of preference preorders, $\{\preccurlyeq_x\}_x$: for each possible query x, the preorder $\preccurlyeq_x$ defines a preference over objects in $\mathcal{Y}$ to be selected. Intuitively, "$y_1 \preccurlyeq_x y_2$" means that y_2 is more relevant to the query x than y_1.

More precisely, for each query $x \in \mathcal{X}$, $\preccurlyeq_x$ is a complete (for each $y_1, y_2 \in \mathcal{Y}$, either $y_1 \preccurlyeq y_2$ or $y_2 \preccurlyeq_x y_1$), reflexive ($y \preccurlyeq_x y$ for all $y \in \mathcal{Y}$), and transitive ($y_1 \preccurlyeq_x y_2$ and $y_2 \preccurlyeq_x y_3$ implies $y_1 \preccurlyeq_x y_3$) relation. For each $x \in \mathcal{X}$, $\preccurlyeq_x$ induces a preordered space $(\mathcal{Y}, \preccurlyeq_x)$. This implicitly defines two additional relations: $\prec_x$ and $\sim_x$. We will write $y_1 \prec_x y_2$ if "$y_1 \preccurlyeq_x y_2$ and not $y_2 \preccurlyeq_x y_1$", and $y_1 \sim y_2$ if "$y_1 \preccurlyeq_x y_2$ and $y_2 \preccurlyeq_x y_1$".

For a collection of objects $\boldsymbol{y} = (y_1, \ldots, y_n) \in \mathcal{Y}^n$ and a query $x \in \mathcal{X}$, the preorder $\preccurlyeq_x$ defines a selection function

$$\text{Select}(x, (y_1, \ldots, y_n)) \coloneqq \max\left((y_1, \ldots, y_n), \texttt{key} = \preccurlyeq_x\right). \tag{9}$$

That is, $\text{Select}(x, \boldsymbol{y})$ returns the most relevant element with respect to the query x. In particular, it returns y_i when $y_i \succ_x y_j$, $\forall j \neq i$ (and may return an arbitrary element if no unique maximal element exists in $(y_1, \ldots, y_n)$).

We will assume some regularity conditions on the family of preorders $\{\preccurlyeq_x\}_x$ which essentially stipulate that: 1) nearby elements in $\mathcal{Y}$ have a similar preference with respect to each x, and 2) nearby queries in $\mathcal{X}$ induce similar preference preorders.

Assumption 1 (Selection criterion is query-continuous and key-continuous)**.** *The family of preorder relations $\{\preccurlyeq_x\}_{x \in \mathcal{X}}$ satisfies the following:*

1. ***Key-continuity.*** *For each $x \in \mathcal{X}$, $\preccurlyeq_x$ is continuous. That is, for any sequence $(y_i)_i$ such that $y_i \preccurlyeq_x z$ and $y_i \to y_\infty$, we have $y_\infty \preccurlyeq_x z$. Equivalently, for any $y \in \mathcal{Y}$, $\{z \in \mathcal{Y} : z \preccurlyeq_x y\}$ and $\{z \in \mathcal{Y} : y \preccurlyeq_x z\}$ are closed sets in $\mathcal{Y}$.*

2. ***Query-continuity.*** *Under key-continuity, Debreu et al. (1954) shows that for each $x \in \mathcal{X}$, there exists a continuous in utility function $u_x : \mathcal{Y} \to \mathbb{R}$ for $\preccurlyeq_x$ such that $y_1 \preccurlyeq_x y_2 \iff u_x(y_1) \leq u_x(y_2)$. For query-continuity, we make the further assumption that there exists a family of utility functions $\{u_x : \mathcal{Y} \to \mathbb{R}\}_{x \in \mathcal{X}}$ such that $u(x, y) \coloneqq u_x(y)$ is also continuous in its first argument.*

For technical reasons, for Equation (9) to make sense, we must assume that there exists a unique element to be selected. We formulate this in terms of an assumption on the data distribution of the space $\mathcal{X} \times \mathcal{Y}^n$. This is a technical assumption, and different forms of such an assumption would be possible (e.g., instead condition on this event).

Assumption 2 (Selection is unique almost always). *Let $(x, \boldsymbol{y}) \sim \mathbb{P}_{x,\boldsymbol{y}}$. For each $\varepsilon > 0$, there exists $\eta_\varepsilon > 0$ such that $\min_{j \neq i} |u_x(y_i) - u_x(y_j)| > \eta_\varepsilon$ with probability at least $1 - \varepsilon$.*

Theorem (Function class of relational attention). *Let $\mathcal{X}, \mathcal{Y}$ be compact Euclidean spaces. Let $\{\preccurlyeq_x\}_{x \in \mathcal{X}}$ be an arbitrary family of relevance preorders on $\mathcal{Y}$ which are query-continuous and key-continuous (Assumption 1). Let $\mathrm{Select}(x, (y_1, \ldots, y_n)) = \max((y_1, \ldots, y_n), \mathrm{key} =\preccurlyeq_x)$ be the selection function associated with $\{\preccurlyeq_x\}_x$. Let $R : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}^{d_r}$ be an arbitrary continuous relation function. Suppose $x, \boldsymbol{y} \sim \mathbb{P}_{x,\boldsymbol{y}}$ and that Assumption 2 holds (i.e., the data distribution is such that there exists a unique most-relevant element). For any $\varepsilon > 0$, there exists multi-layer perceptrons $\phi_q^{\mathrm{attn}}, \phi_k^{\mathrm{attn}}, \phi_q^{\mathrm{rel}}, \phi_k^{\mathrm{rel}}$ and a choice of symbols such that,*

$$\|\mathrm{RA}(x, (y_1, \ldots, y_n)) - R(x, \mathrm{Select}(x, (y_1, \ldots, y_n)))\|_\infty < \varepsilon$$

Proof. Condition on the event $\mathcal{E} := \{(x, \boldsymbol{y}) \in \mathcal{X} \times \mathcal{Y}^n : \min_{j \neq i} |u_x(y_i) - u_x(y_j)| > \eta_\varepsilon\}$. Let $i^* = \arg\max((y_1, \ldots, y_n), \mathrm{key} =\preccurlyeq_x) = \arg\max(u_x(y_1), \ldots, u_x(y_n))$. By (Altabaa and Lafferty, 2024a, Theorem 5.1), for any $\varepsilon_1 > 0$, there exists MLPs $\phi_q^{\mathrm{attn}}, \phi_k^{\mathrm{attn}}$ such that $\alpha_{i^*}(x, \boldsymbol{y}) > 1 - \varepsilon_1$ for any $(x, \boldsymbol{y}) \in \mathcal{E}$. That is, the attention score is nearly 1 for the $\preccurlyeq_x$-selected element *uniformly* over inputs in $\mathcal{E}$.

Similarly, by (Altabaa and Lafferty, 2024a, Theorem 3.1), for any $\varepsilon_2 > 0$, there exists MLPs $(\phi_{q,\ell}^{\mathrm{rel}}, \phi_{k,\ell}^{\mathrm{rel}})_{\ell \in [d_r]}$ such that $r(x, y) := (\langle \phi_{q,\ell}^{\mathrm{rel}}(x), \phi_{k,\ell}^{\mathrm{rel}}(y) \rangle)_{\ell \in [d_r]}$ approximates the target relation R uniformly within an error of ε_2,

$$\|R(x, y) - r(x, y)\|_\infty < \varepsilon_2, \quad \text{Lebesgue almost every } (x, y) \in \mathcal{X} \times \mathcal{Y}.$$

Thus, we have

$$\|\mathrm{RA}(x, (y_1, \ldots, y_n)) - R(x, \mathrm{Select}(x, (y_1, \ldots, y_n)))\|_\infty$$
$$= \left\| \sum_{i=1}^n \alpha_i(x; \boldsymbol{y})\, r(x, y_i) - R(x, y_{i^*}) \right\|_\infty$$
$$\leq \sum_{i=1}^n \|\alpha_i(x; \boldsymbol{y})\, r(x, y_i) - R(x, y_{i^*})\|_\infty$$
$$\leq \alpha_{i^*}(x, \boldsymbol{y})\, \|r(x, y_{i^*}) - R(x, y_{i^*})\|_\infty + \sum_{j \neq i^*} \alpha_i(x; \boldsymbol{y})\, \|r(x, y_i) - R(x, y_{i^*})\|_\infty$$
$$\leq (1 - \varepsilon_1)\varepsilon_2 + \varepsilon_1 \max_{x,y,y^*} \|r(x, y) - R(x, y^*)\|_\infty.$$

Note that $\max_{x,y,y^*} \|r(x, y) - R(x, y^*)\|_\infty$ is finite since $\mathcal{X}, \mathcal{Y}$ are compact and r, R are continuous. Letting $\varepsilon_1, \varepsilon_2$ be small enough completes the proof. $\qquad\square$

To summarize the analysis in this section, we showed that relational attention can approximate any computation composed of selecting an object from a collection followed by computing a relation with that object. We can approximate any well-behaved selection criterion by formulating it in terms of an abstract preference preorder, and approximating the corresponding utility function (given by a Debreu representation theorem) by inner products of query and key feature maps. We can then approximate the target relation function similarly by inner products of a different set of query and key feature maps.

In the analysis above, we set aside the role of the symbols. Note that the function class this approximation result proves involves retrieving a relation from a selected object, but does not explicitly encode the identity of the selected object. Informally, the receiver knows that it has a particular relation with one of the objects in its context, and knows that this relation is with an object that was selected according to a particular selection criterion, but does not know the identity of the object beyond that. This is the purpose of adding symbols to relational attention—the retrieved relation is tagged with a symbol identifying the sender.

B Architecture & Implementation Details

In this section, we briefly discuss some details of implementation that may be of interest to the reader. Our code is publicly available through the project git repository and includes detailed instructions for reproducing our experimental results. We also make available detailed experimental logs for each experimental run which include: the git commit ID giving the version of the code that was used to run the experiment, the exact command and script used to run the script, the hardware used for that run, and metrics tracked over the course of training. Our code uses the PyTorch framework.

B.1 Relational Attention and Dual-Head Attention

The relational attention operation is defined as part of dual-head attention in Algorithm 1. We briefly mention some details of the implementation. Let $n_h = n_h^{sa} + n_h^{ra}$ be the total number of parameters The learnable parameters here are $W_q^h, W_k^h \in \mathbb{R}^{d_{\text{model}} \times d_{\text{key}}}$, $W_v^h \in \mathbb{R}^{d_{\text{model}} \times d_h}$, $h \in [n_h^{sa}], W_o^{sa} \in \mathbb{R}^{d_{\text{model}} \times d_{\text{model}}}$, for the self-attention heads, and $W_{q,h}^{\text{attn}}, W_{k,h}^{\text{attn}} \in \mathbb{R}^{d_{\text{model}} \times d_{\text{key}}}, W_s^h \in \mathbb{R}^{d_{\text{model}} \times d_h}, W_r^h \in \mathbb{R}^{d_r \times d_h}, h \in [n_h^{ra}], W_{q,\ell}^{\text{rel}}, W_{k,\ell}^{\text{rel}} \in \mathbb{R}^{d_{\text{model}} \times d_{\text{proj}}}, \ell \in [d_r], W_o^{ra} \in \mathbb{R}^{d_{\text{model}} \times d_{\text{model}}}$ for the relational attention heads. We let $d_{\text{key}}, d_h = d_{\text{model}}/n_h$ to maintain the same dimension for the input and output objects. By default, and for all our experiments, we let $d_r = n_h^{ra}$ and $d_{\text{proj}} = d_{\text{key}}$. To model symmetric relations, we let $W_{q,\ell}^{\text{rel}} = W_{k,\ell}^{\text{rel}}$.

Note that the same d_r-dimensional relation is used for all n_h^{ra} attention heads, with a learned linear map W_r^h for each head extracting the relevant aspects of the relation for that attention head. This allows for useful computations to be shared across all heads. Note also that the head dimension $d_h = d_{\text{model}}/n_h$ is defined in terms of the total number of attention heads and is the same per-head for both self-attention and relational attention. This means that the number of components in the d_{model}-dimensional output corresponding to each attention head type is proportional to the number of heads of that type. For example, if $n_h^{sa} = 6, n_h^{ra} = 2$, then 75% of the d_{model}-dimensional output is composed of the output of self-attention heads and 25% is composed of the output of relational attention heads. This enables tuning the relative importance of each head type for the task.

Finally, we note that relational attention can be implemented with an `einsum` operation. For example, in PyTorch, it looks something like the following.

```
# sv: (bs, seqlen, n_heads, head_dim)
# attn_scores: (bs, n_heads, seqlen, seqlen)
# relations: (bs, seqlen, seqlen, n_heads, head_dim)

attended_symbols = torch.einsum('bhij,bjhd->bihd', attn_scores, sv)
# shape: (bs, seqlen, n_heads, head_dim)
attended_relations = torch.einsum('bhij,bijhd->bihd', attn_scores,
                                   proj_relations)
# shape: (bs, seqlen, n_heads, head_dim)
output = attended_symbols + attended_relations
# shape: (bs, seqlen, n_heads, head_dim)
```

Here, `sv[:,:,h,:]` $= s W_s^h$, `attn_scores[:,h,:,:]` $= \alpha^h$, and `relations[:,:,h,:]` $= r W_r^h$, which can all be computed with simple matrix multiplication operations. When using position-relative symbols, the first line is replaced with `attended_symbols = torch.einsum('bhij,ijhd->bihd', attn_scores, sv)`, assuming that `sv[i,j,h,:]` $= s_{j-i} W_s^h$

Weight-tying self-attention and relational-attention heads. To increase parameter efficiency and perhaps improve performance, we suggest weight-tying the attention query/key projections across self-attention and relational attention heads. The intuition is that the same selection criteria may be useful for retrieving either sensory or relational information from a particular object. By sharing these projections across heads, we can reduce the number of trainable parameters, and introduce an additional supervisory signal which may improve performance. We leave testing this idea for future work.

Composing relational attention. We remark that composing relational attention modules can be interpreted as representing hierarchical or higher-order relations. That is, relations between relations. An example of this is the relation tested in the `match pattern` task in the relational games benchmark. After one iteration of relational attention, an object's representation is updated

with the relations it has with its context. A second iteration of relational attention now computes a representation of the relation between an object's relations and the relations of the objects in its context.

B.2   Encoder and Decoder Blocks

We briefly mention a few configurations in our implementation that appear in our experiments. We aimed to make our implementation configurable to allow for various tweaks and optimizations that have been found in the literature for training Transformer models.

Symbol assignment. A shared symbol assignment module is used for all layers in the model. We explore three types of symbol assignment mechanisms: positional symbols, position-relative symbols, and symbolic attention. Different symbol assignment mechanisms are more well-suited to different tasks. We discuss ablation experiments we carried out on the effect of the symbol assignment mechanism in Appendix C.

MLP block. The MLP block uses a 2-layer feedforward network with a configurable activation function. The intermediate layer size is $d_{\text{ff}} = 4 \cdot d_{\text{model}}$ by default. We also use the SwiGLU "activation function" (Shazeer, 2020) in some of our experiments. SwiGLU is not merely an activation function, but is rather a neural network layer defined as the component-wise product of two linear transformations of the input. It is a type of gated linear unit (Dauphin et al., 2017) with the sigmoid activation replaced with a Swish activation (Hendrycks and Gimpel, 2016), $\text{SwiGLU}(x) = \text{Swish}(xW + b) \otimes (xV + c)$. This is used in the Llama series of models and was found to be a useful modification (Touvron et al., 2023).

Normalization. Either LayerNorm (Ba et al., 2016) or RMSNorm (Zhang and Sennrich, 2019) can be used. Normalization can be performed post-attention, like in the original Transformer paper (Vaswani et al., 2017), or pre-attention as in (Xiong et al., 2020).

Positional encoding. Our experiments use either learned positional embeddings or RoPE (Su et al., 2023).

C   Experimental Details & Further Discussion

C.1   Relational Games (Section 4.1)

Experimental details

Dataset details. The relational games benchmark datasets consists of $36 \times 36 \times 3$ RGB images depicting a 3×3 grid of objects which satisfy a particular visual relationship. The set of objects consists of simple geometric shapes. For example, in the `occurs` task, one object is present in the top row and three in the bottom row, and the task is to determine whether the object in the top row occurs (i.e., is among) the objects in the bottom row. The most difficult task in the benchmark is the `match pattern` task, where the grid contains a triplet of objects in the top row and another triplet of objects in the bottom row. Each triplet satisfies some relationship (e.g., ABC, ABA, ABB, or AAB), and the task is to determine whether the relation in the first triplet is the same as the relation in the second triplet. The difficulty in solving this task is that it requires parsing a second-order relation (a relation between relations). We remark that composing relational attention modules naturally captures this kind of hierarchical relations: one relational attention operation produces objects with a relational representation and another would compute relations between those relations.

Model architectures. We use a Vision-Transformer-type architecture where the input image is split up into patches, flattened, and passed through the sequence model with added learned positional embeddings. We use average pooling at the end and pass through an MLP to produce the final prediction. We use a patch size of 12×12 which separates objects according to the grid structure. We note that in more general visual relational reasoning tasks where there isn't this type of grid structure, it would be appropriate to combine our approach with an object-discovery module such as Slot Attention (Locatello et al., 2020).

We use the following hyperparameters: 2 layers, $d_{\text{model}} = 128$, $d_{\text{ff}} = 256$, SwiGLU "activation", dropout rate = 0.1, and pre-LayerNormalization. For the *DAT* models, we use positional symbols as the symbol assignment mechanism. The total number of heads is 2. For the Transformer model, there are only self-attention heads: $n_h^{sa} = 2$. For *DAT*, we evaluated two configurations for the composition of head types, one with $n_h^{sa} = n_h^{ra} = 1$ and one with only relational attention heads

$n_h^{sa} = 0, n_h^{ra} = 2$. We also evaluated variants with and without the constraint that the relations in relational attention are symmetric (i.e., $W_q^{\mathrm{rel}} = W_k^{\mathrm{rel}}$).

Training details. For each task and model, we evaluated learning curves by varying the training set size and training the model until convergence, then evaluating on a hold-out test set. For four out of five of the tasks, we evaluate learning curves within the range of 250 to 2,500 samples, in increments of 250. For the more difficult `match pattern`, the range is from 5,000 to 25,000 in increments of 5,000. The ranges were chosen based on the difficulty of the different tasks in order to identify the right "resolution". When evaluating learning curves, each training set is sampled randomly from the full dataset. For each task, model, and training set size, we repeat the experiment 5 times with different random seeds to compute approximate confidence intervals (accounting for randomness in sampling the dataset and random initialization). We use an Adam optimizer with a learning rate of 0.001, $\beta_1 = 0.9$, $\beta_2 = 0.99$, and a batch size of 512. We train for 50 epochs.

Further Discussion, Exploration, & Ablations

We performed an ablation over the *symmetry* inductive bias in the relations computed in relational attention. Our implementation exposes an argument which controls whether the relation $r(x, y) = (\langle W_{q,\ell}^{\mathrm{rel}}, W_{k,\ell}^{\mathrm{rel}} \rangle)_{\ell \in [d_r]} \in \mathbb{R}^{d_r}$ modeled in relational attention is constrained to be symmetric by setting $W_{q,\ell}^{\mathrm{rel}} = W_{k,\ell}^{\mathrm{rel}}$. Indeed, we find symmetry to be a useful inductive bias in this task. Figure 6 depicts learning curves for the two configurations of *DAT* comparing symmetric RA against asymmetric RA. We find that symmetry results in faster learning curves for both configurations.
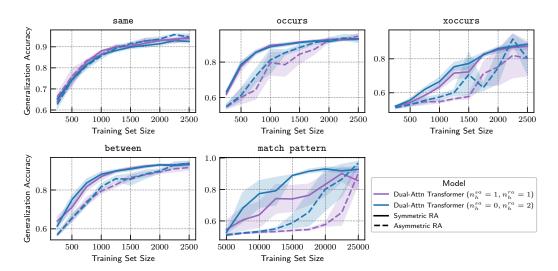

Figure 6: An ablation of the effect of symmetry in relational attention in the relational games experiments.

C.2 Mathematical Problem-Solving (Section 4.2)

Experimental details

Dataset details. Saxton et al. (2019) propose a benchmark to assess neural models' ability to perform mathematical reasoning. The dataset consists of a suite of tasks in free-form textual input/output format. The tasks cover several topics in mathematics, including arithmetic, algebra, and calculus. For each task, the authors programmatically generate 2×10^6 training examples and 10^4 validation examples. Questions have a maximum length of 160 characters and answers have a maximum length of 30 characters.

Model architectures. We use an encoder-decoder architecture for this experiment, treating it as a sequence-to-sequence task. We use character-level encoding with a common alphabet of size 85 containing small and upper case letters, digits 0-9, and symbols (e.g., `*`, `/`, `+`, `-`). The models have 2-layer encoders and decoders, with $d_{\mathrm{model}} = 128, d_{\mathrm{ff}} = 256$, ReLU activation, dropout rate = 0.1, and post-normalization. Sinusoidal positional embeddings are used as the positional encoding method.
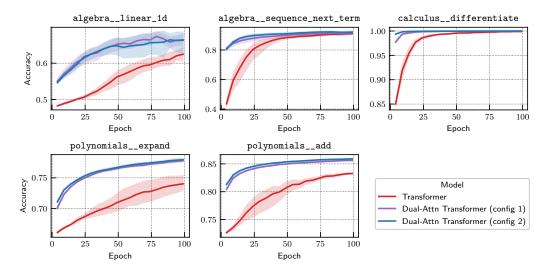
Figure 7: Extrapolation validation accuracy over the course of training for mathematical problem-solving tasks.

For all models, the total number of attention heads (across self-attention and relational attention) is 8. For the Transformer model, there are only self-attention heads: $n_h^{sa} = 8$ for both the encoder and decoder. For *DAT*, we evaluated two configurations for the composition of head types, one with $n_h^{sa} = n_h^{ra} = 4$ in the encoder and $n_h^{sa} = 8, n_h^{ra} = 0$ in the decoder (i.e., standard Transformer Decoder), and one with $n_h^{sa} = 4 = n_h^{ra} = 4$ in the encoder and $n_h^{sa} = 4 = n_h^{ra} = 4$ in the decoder. The number of cross-attention heads in the decoder is 8 in all cases. No symmetry constraint is made on relational attention. Position-relative symbols are used as the symbol assignment mechanism, and the symbol library is shared across all layers in both the encoder and decoder.

Training Details. Each model is trained on each task for 50 epochs. We use the Adam optimizer with $\beta_1 = 0.9, \beta_2 = 0.995$, a learning rate of 6×10^{-4}, and a batch size of 128. We evaluate and track the per-character accuracy over the course of training. We repeat this process 5 times for each combination of model and task with different random seeds to compute approximate confidence intervals.

Further Discussion, Exploration, & Ablations

This benchmark contains two validation splits: an interpolation split and an extrapolation split. The interpolation split is generated by the same procedure as the training split. The extrapolation split is generated with different parameters to make the task more difficult. We refer to the original reference for more details (Saxton et al., 2019). Figure 3 in the main text depicts the interpolation validation accuracy. Figure 7 depicts the extrapolation validation accuracy over the course of training. For completeness, we also show the training accuracy in Figure 8.

C.3 Language Modeling (Section 4.3)

Experimental details

Dataset details. The Tiny Stories dataset by Eldan and Li (2023) is a language modeling benchmark designed to evaluate small language models. The dataset consists of short stories and is synthetically generated by GPT-3.5 and GPT-4.

Model architectures. We use a Decoder-only architecture, with causal attention for autoregressive language modeling. We fix the total number of attention heads to be 8 for all models. For the Transformer model, there are only self-attention heads: $n_h^{sa} = 2$. For *DAT*, we evaluated two configurations for the composition of head types, one with $n_h^{sa} = 6, n_h^{ra} = 2$ (more self-attention heads) and one with $n_h^{sa} = n_h^{ra} = 4$ (a balanced composition of head types). We experiment with $d_{\text{model}} \in \{64, 128\}$ and number of layers $L \in \{4, 5, 6\}$. For all models, we use $d_{\text{ff}} = 4d_{\text{model}}$, SwiGLU activation in the MLP, RoPE positional encoding, no bias, dropout rate = 0.1, and pre-LayerNormalization. We use weight-tying between the embedding layer and the final prediction
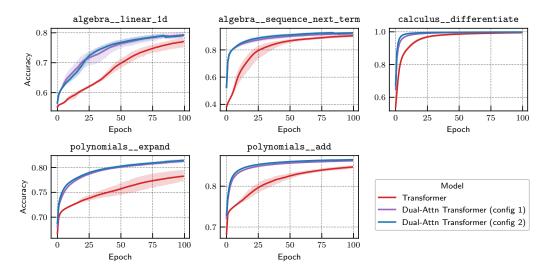
Figure 8: Training accuracy over the course of training for mathematical problem-solving tasks.
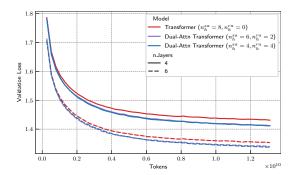

Figure 9: Loss curves of *DAT* language models with $d_{\mathrm{model}} = 128$.

layer (Inan et al., 2017). For *DAT* models, we perform ablations on the effect of the choice of symbol assignment mechanism and symmetry in relational attention. We train models with position-relative symbols and symbolic attention, as well as models with symmetric and asymmetric relations in relational attention. We use a context size of 512 for all models. Text is tokenized using the Llama SentencePiece tokenizer (Touvron et al., 2023), which contains $32,000$ tokens.

Figure 4 in the main text depicts the $d_{\mathrm{model}} = 64$ models with varying number of layers $L \in \{4, 5, 6\}$. Figure 9 shows the loss curves for larger models with $d_{\mathrm{model}} = 128$. The *DAT* models use symbolic attention as the symbol assignment mechanism and asymmetric relations in relational attention. Symbolic attention uses $n_s = \texttt{context\_length} = 512$ symbols with 4 heads.

Training Details. All models are trained with the AdamW optimizer with a constant learning rate of 0.001 and $\beta_1 = 0.9, \beta_2 = 0.95$. We clip gradients to a norm of 1. Models are trained for $100,000$ iterations. We use a batch size of 128 with 2 gradient accumulation steps. Thus, each step trains on $131,072$ tokens.
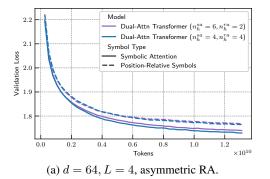
Further Discussion, Exploration, & Ablations

Table 1 shows the end-of-training validation loss and perplexity for all models we evaluated. For each d_{model}, L the best-performing model is bolded. We find that the *DAT* model with asymmetric relational attention and symbolic attention consistently performs the best, beating out the Transformer by a small margin.

Figure 10 depicts an ablation over symbol type. For $d_{\mathrm{model}} = 64, L = 4$, it shows training curves for *DAT* models with two types of symbol retrieval mechanism: symbolic attention and position-relative symbol. We find that the symbolic attention variant learns faster and reaches a smaller loss. This holds

Table 1: Language Modeling on the Tiny Stories Dataset

d_{model}	L	n_h^{sa}	n_h^{ra}	Symbol Assignment Mechanism	Symmetric RA	val/loss	val/perplexity
64	4	4	4	Position-Relative Symbols	False	1.764	5.840
					True	1.785	5.963
				Symbolic Attention	False	**1.729**	**5.639**
					True	1.744	5.722
		6	2	Position-Relative Symbols	False	1.768	5.859
					True	1.777	5.914
				Symbolic Attention	False	1.740	5.697
					True	1.745	5.727
		8	0	NA	NA	1.775	5.903
	5	4	4	Symbolic Attention	False	**1.692**	**5.431**
					True	1.698	5.467
		6	2	Position-Relative Symbols	True	1.730	5.640
				Symbolic Attention	False	**1.692**	**5.432**
					True	1.704	5.495
		8	0	NA	NA	1.730	5.640
	6	4	4	Position-Relative Symbols	False	1.685	5.395
					True	1.704	5.498
				Symbolic Attention	False	**1.656**	**5.239**
					True	1.668	5.303
		6	2	Position-Relative Symbols	True	1.691	5.424
				Symbolic Attention	False	1.663	5.277
					True	1.669	5.308
		8	0	NA	NA	1.692	5.431
128	4	4	4	Symbolic Attention	False	**1.411**	**4.102**
					True	1.417	4.127
		6	2	Symbolic Attention	False	1.412	4.105
					True	1.415	4.118
		8	0	NA	NA	1.431	4.183
	6	4	4	Symbolic Attention	False	**1.337**	**3.811**
					True	1.346	3.843
		6	2	Symbolic Attention	False	1.340	**3.818**
					True	1.346	3.843
		8	0	NA	NA	1.353	3.870

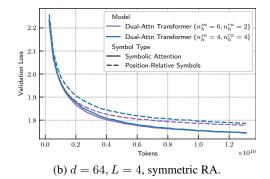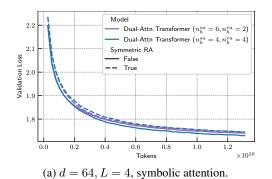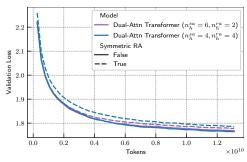(a) $d = 64$, $L = 4$, asymmetric RA.

(b) $d = 64$, $L = 4$, symmetric RA.

Figure 10: Ablation of symbol assignment mechanism. Symbolic attention outperforms position-relative symbols.

(a) $d = 64, L = 4$, symbolic attention.

(b) $d = 64, L = 4$, position-relative symbol assignment

Figure 11: Ablation of Symmetry in relational attention. The *DAT* models with asymmetric relations perform slightly better than those with a symmetry constraint.

both when relational attention uses symmetric or asymmetric relations. We posit the explanation that the differentiable equivalence class mapping implemented by symbolic attention may capture a form of syntax.

Figure 11 shows an ablation over the symmetry of relational attention. Relational attention can be constrained to represent symmetric relations by imposing that $W_q^{\text{rel}} = W_k^{\text{rel}}$. This is sometimes a useful inductive bias, as observed in the relational games experiments. However, in the language modeling experiments, we found that it was not a useful inductive bias and that asymmetric relational attention outperformed symmetric relational attention. One possible explanation is that the types of relations that are relevant in parsing language are in general asymmetric. For example, syntactic or grammatical relations such as noun-verb, subject-object, determiner-noun, etc. are asymmetric.

C.4   Image Recognition (Section 4.4)

Experimental details

Dataset details. For this experiment, we use the ImageNet Large Scale Visual Recognition Challenge 2012 (ILSVRC2012) dataset (Russakovsky et al., 2015). The dataset is roughly 140GB in size and consists of RGB images hand-labeled with the presence or absence of 1,000 object categories. We train on images resized to 224×224. In the training set, we perform a random crop and resize to 224×224, then normalize channel-wise with means $(0.485, 0.456, 0.406)$ and standard deviations $(0.229, 0.224, 0.225)$. For the validation set, we resize to 256×256 then center crop to 224×224, and normalize in the same way.

Model architectures. We use a Vision Transformer-style architecture (Dosovitskiy et al., 2021). ImageNet's $224 \times 224 \times 3$ RGB images are divided into 16×16 patches, flattened, and linearly embedded into a vector. A learnable positional embedding is added to each patch embedding. We also prepend a special classification token. The sequence of patch embeddings is then fed through an Encoder and the embedding of the class token is used to generate the final classification through a fully connected layer. We compare a Vision Transformer model with $n_h^{sa} = 16$ to an *DAT* model with $n_h^{sa} = 10, n_h^{sa} = 6$. For both, we used a model dimension $d_{\text{model}} = 1024$, $L = 24$ layers, MLP hidden dimension $d_{\text{ff}} = 4096$, SwiGLU activation, no bias, dropout rate = 0.1, and pre-LayerNormalization. The *DAT* model uses position-relative symbols as the symbol assignment mechanism and symmetric relational attention.

Training details. Both models are trained with the AdamW optimizer, with a constant learning rate of 5×10^{-4} and $\beta_1 = 0.9, \beta_2 = 0.99$. We used a batch size of 32 and 32 gradient accumulation steps.

D   Comparison to Altabaa et al. (2024): Abstractors and Relational Cross-Attention

A closely related work is Altabaa et al. (2024), which proposes a Transformer-based module called the "Abstractor" with relational inductive biases. The core operation in the Abstractor is a variant

of attention dubbed "relational cross-attention" (RCA). In this section, we will discuss the relation between the Dual Attention Transformer and the Abstractor.

D.1 Comparison between RA (this work) and RCA (Altabaa et al., 2024)

Altabaa et al. (2024) propose a variant of attention called relational cross-attention which shares some characteristics with our proposal of what we're calling "relational attention" in this work. In this discussion, we will use the acronyms RCA and RA, respectively to distinguish between the two.

RCA processes a sequence of objects $\boldsymbol{x} = (x_1, \ldots, x_n)$ and produces a sequence of objects $\boldsymbol{x}' = (x'_1, \ldots, x'_n)$ via the following operation

$$\boldsymbol{x}' \leftarrow \sigma_{\mathrm{rel}}\left(\phi_q(\boldsymbol{x})\phi_k(\boldsymbol{x})^{\mathsf{T}}\right)\boldsymbol{s},$$
$$\boldsymbol{s} = \mathrm{SymbolRetriever}(\boldsymbol{x})$$

where ϕ_q, ϕ_k are query and key transformations, and the symbols $\boldsymbol{s}$ take the same role as in this work. σ_{rel} is referred to as a "relation activation". It may be either softmax or an element-wise activation (e.g., tanh, sigmoid, or linear). For the purposes of this discussion, let us consider $\sigma_{\mathrm{rel}} = \mathrm{Softmax}$, which was used in the majority of the experiments in (Altabaa et al., 2024).

To facilitate the discussion, let us write RA and RCA side-by-side using a common notation.

RA (this work)	RCA (Altabaa et al., 2024)
$(x'_1, \ldots, x'_n) \leftarrow \mathrm{RA}(\boldsymbol{x}; S_{\mathrm{lib}}),$	$(x'_1, \ldots, x'_n) \leftarrow \mathrm{RCA}(\boldsymbol{x}; S_{\mathrm{lib}})$
$x'_i = \sum_{j=1}^{n} \alpha_{ij}\left(W_r\, r(x_i, x_j) + W_s\, s_j\right),$	$x'_i = \sum_{j=1}^{n} \alpha_{ij}\, s_j,$
$\boldsymbol{\alpha} = \mathrm{Softmax}\left(\phi_q(\boldsymbol{x})\phi_k(\boldsymbol{x})^{\mathsf{T}}\right),$	$\boldsymbol{\alpha} = \mathrm{Softmax}\left(\phi_q(\boldsymbol{x})\phi_k(\boldsymbol{x})^{\mathsf{T}}\right),$
$r(x, y) = \left(\left\langle \phi_{q,\ell}^{\mathrm{rel}}(x), \phi_{k,\ell}^{\mathrm{rel}}(y)\right\rangle\right)_{\ell \in [d_r]},$	
$(s_1, \ldots, s_n) = \mathrm{SymbolRetriever}(\boldsymbol{x};\, S_{\mathrm{lib}})$	$(s_1, \ldots, s_n) = \mathrm{SymbolRetriever}(\boldsymbol{x};\, S_{\mathrm{lib}})$

RCA can be understood as self-attention, but the values are replaced with symbols (i.e., $\mathrm{Attention}(Q \leftarrow \boldsymbol{x},\ K \leftarrow \boldsymbol{x},\ V \leftarrow \boldsymbol{s})$). By viewing the attention scores α_{ij} as relations, this has the effect of producing a relation-centric representation. The rationale is that in standard self-attention, the attention scores form a type of relation, but these relations are only used in an intermediate processing step in an information-retrieval operation. The relations encoded in the attention scores are entangled with the object-level features, which have much greater variability. This thinking also motivates the design of RA in the present work.

RCA can be understood as computing a pairwise relation $\langle \phi_q^{\mathrm{attn}}(x_i), \phi_k^{\mathrm{attn}}(x_j)\rangle$ between x_i and each x_j in the context, and retrieving the symbol s_j associated with the object x_j with which the relation is strongest. That is, RCA treats the relations and the attention scores as the same thing. By contrast, the attention operation and computation of relations are disentangled in RA. The attention component is modeled by one set of query/key maps $\phi_q^{\mathrm{attn}}, \phi_k^{\mathrm{attn}}$ and the relation component is modeled by another set of query/key maps $(\phi_{q,\ell}^{\mathrm{rel}}, \phi_{k,\ell}^{\mathrm{rel}})_{\ell \in [d_r]}$.

The intuitive reason for this choice is that, for many tasks, the optimal "selection criterion" will be different from the task-relevant relation. For example, in a language modeling task, you may want to attend to objects on the basis of proximity and/or syntax while being interested in a relation based on semantics. Similarly, in a vision task, you may want to attend to objects on the basis of proximity, while computing a relation across a certain visual attribute.

In RA, the symbols maintain the role of identifying the sender. But instead of being the whole message, they are attached to a relation.

D.2 Comparison between *DAT* and the Abstractor

We now briefly discuss the differences in the resulting architectures. Altabaa et al. (2024) propose an encoder-like module called the Abstractor which consists of essentially replacing self-attention in an Encoder with relational cross-attention. That is, it consists of iteratively performing RCA followed by

an MLP. The paper proposes several ways to incorporate this into the broader Transformer architecture. For example, some of the experiments use a Encoder → Abstractor → Decoder architecture to perform a sequence-to-sequence task. Here, the output of a standard Transformer is fed into an Abstractor, and the Decoder cross-attends to the output of the Abstractor. In another sequence-to-sequence experiment, Altabaa et al. (2024) use an architecture where the Decoder cross-attends to both the Encoder and the Abstractor, making use of both sensory and relational information. In particular, the standard encoder and decoder blocks are the same (focusing on sensory information), but an additional module is inserted in between with a relational inductive bias.

By contrast, our approach in this paper is to propose novel encoder and decoder architectures imbued with two distinct types of attention heads, one with an inductive bias for sensory information and the other with an inductive bias for relational information. This has several potential advantages. The first is versatility and generality. The Abstractor architectures that were explored in (Altabaa et al., 2024) only explicitly support sequence-to-sequence or discriminative tasks. For example, they do not support autoregressive models like modern decoder-only language models (e.g., of the form we experiment with in Section 4.3). Moreover, even in sequence-to-sequence tasks, Abstractor architectures only support relational processing over the input sequence, but they do not support relational processing over the target sequence (since the decoder does not have RCA). Another potential advantage of *DAT* is simplicity. The Abstractor paper proposes several architectures and configurations for the Encoder/Abstractor/Decoder modules, introducing several hyperparameters that are not trivial to choose. Moreover, it is unclear how to interpret this kind of architecture as the number of layers increases, and the original paper does not experiment with scaling up the number of layers. The final potential advantage is increased expressivity. In *DAT*, the two types of attention heads exist side by side in each layer. This allows relational attention heads to attend to the output of the self-attention heads at the previous layer, and vice-versa. This yields broader representational capacity, and potentially more interesting behavior as we scale the number of layers.

D.3   How would RCA perform in an *DAT*-style dual head-type architecture?

One question one might ask is: how would an *DAT*-style dual head-type architecture perform if we used Altabaa et al. (2024)'s RCA instead of the RA head-type proposed in this work? We carried out a few ablation experiments to answer this question.

Figure 12 compares learning curves on the relational games benchmark between standard *DAT* (with RA-heads) and a version of *DAT* with Altabaa et al. (2024)'s RCA heads. We find that the two models perform similarly, with most differences small enough to be within the margin of error. This figure depicts the configuration with asymmetric RA and positional symbols.

Figure 13 depicts the validation loss curves on the Tiny Stories language modeling task, comparing standard *DAT* against a version with RCA heads. Here, we find that our relational attention heads yield better-performing models, with the RCA-head variant of *DAT* performing no better than a standard Transformer with a matching total number of heads.
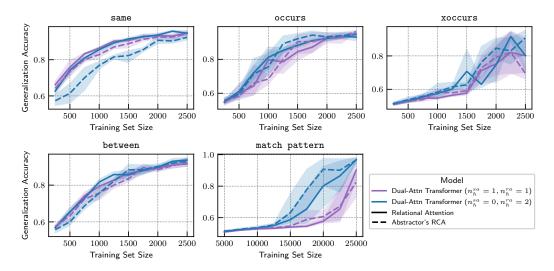
Figure 12: Learning curves for *DAT* with RA compared with *DAT* with RCA on the relational games benchmark. The performance is similar, with most differences within the margin of error.
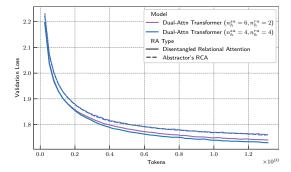

Figure 13: Ablation of relational attention type. The solid line depicts the form of relational attention proposed in this work. The dotted line depicts RCA as proposed by Altabaa et al. (2024). We find that our relational attention mechanism performs better, whereas RCA performs no better than a Transformer.