

449 **Appendices**

450	A Method Details	15
451	A.1 Fourier Conversions	15
452	A.2 Fourier Extraction	15
453	A.3 Sparse Fourier Optimization	16
454	B Experimental Details	17
455	B.1 Implementation Details	17
456	B.1.1 Hyper-parameters	17
457	B.1.2 Sentiment Analysis	17
458	B.1.3 HotpotQA	17
459	B.1.4 DROP	17
460	B.1.5 MS-COCO	18
461	B.2 Measuring Spectral Hierarchies	18
462	B.3 Sparsification	18
463	B.4 Proxy Model Selection	19
464	B.5 Practical Implications	20
465	C Case Study Details	22
466	C.1 Data Attribution via Non-Linear Datamodels	22
467	C.2 Model Component Attribution	25

468 A Method Details

469 A.1 Fourier Conversions

INTERACTION INDEX	FOURIER CONVERSION
Banzhaf ψ_i	$-2F(\{i\})$
Shapley ϕ_i	$(-2) \sum_{\substack{S \supseteq \{i\} \\ S \text{ is odd}}} \frac{F(S)}{ S }$
Influence ξ_i	$\sum_{S \ni i} F(S)^2$
Möbius $I^M(T)$	$(-2)^{ T } \sum_{S \supseteq T} F(S)$
Or $I^O(T)$	$\begin{cases} \sum_{S \subseteq [n]} F(S) & \text{if } T = \emptyset \\ -(-2)^{ T } \sum_{S \supseteq T} (-1)^{ S } F(S) & \text{if } T \neq \emptyset \end{cases}$
Banzhaf Interaction $I^B(T)$	$-2F(T)$
Shapley Interaction $I^S(T)$	$(-2)^{ T } \sum_{S \supseteq T \text{ s.t. } (-1)^{ S } = (-1)^{ T }} \frac{F(S)}{ S - T + 1}$
Shapley Taylor $I_\ell^{ST}(T)$	$\begin{cases} I^M(T), & T < \ell, \\ \sum_{S \supseteq T} \binom{ S }{\ell}^{-1} I^M(S), & T = \ell. \end{cases}$
Faith-Banzhaf $I_\ell^{FB}(T)$	$(-2)^{ T } \sum_{\substack{S \supseteq T \\ S \leq \ell}} F(S)$
Faith-Shapley $I_\ell^{FS}(T)$	$I^M(T) + (-1)^{\ell - T } \frac{ T }{\ell + T } \binom{\ell}{ T } \sum_{\substack{S \supseteq T \\ S > \ell}} F(S) \gamma(S, T, \ell)$ where $\gamma(S, T, \ell) = \sum_{\substack{T \subset R \subset S \\ R > \ell}} \frac{\binom{ R - 1}{\ell + T }}{\binom{ R + \ell - 1}{\ell + T }} (-2)^{ R }$

470 The relationship between Fourier coefficients and influence scores are provided in [36]. We derive the
471 conversion between Fourier and the OR interaction index [58] in this work. All remaining conversions
472 are derived in Appendix C of [4].

473 A.2 Fourier Extraction

474 The exact Fourier transform of a decision tree can be computed recursively [5, 59, 60]. Due to
475 the linearity of the Fourier transform, the Fourier transform of each boosted tree can be computed
476 separately and added together. Algorithm 1, provided by [5], proceeds by traversing the nodes of
477 each tree and summing the resultant Fourier transforms.

Algorithm 1 Fourier Extraction from Gradient Boosted Trees [5]

Require: Gradient boosted model \mathcal{M}

Ensure: Fourier mapping \mathcal{F}

```
1: Initialize  $\mathcal{F} \leftarrow \emptyset$ 
2: for Tree  $T$  in  $\mathcal{M}$  do
3:    $\mathcal{F} \leftarrow \mathcal{F}.\text{merge}(\text{EXTRACTTREE}(T.\text{root}))$  ▷ Add mappings of the individual trees
4: end for
5: return  $\mathcal{F}$ 

6: procedure EXTRACTTREE(node  $n$ )
7:   if  $n$  is leaf then
8:     return  $\{\emptyset \mapsto n.\text{value}\}$ 
9:   else
10:     $\mathcal{N}_L \leftarrow \text{EXTRACTTREE}(n.\text{leftChild})$ 
11:     $\mathcal{N}_R \leftarrow \text{EXTRACTTREE}(n.\text{rightChild})$ 
12:     $\mathcal{N} \leftarrow \emptyset$ 
13:    for  $S$  in  $(\mathcal{N}_L.\text{keys} \cup \mathcal{N}_R.\text{keys})$  do ▷ Mapping returns 0 if not contained
14:       $v_L \leftarrow \mathcal{N}_L[S]$ 
15:       $v_R \leftarrow \mathcal{N}_R[S]$ 
16:       $\mathcal{N}[S] \leftarrow (v_L + v_R)/2$ 
17:       $\mathcal{N}[S \cup \{n.\text{featureSplit}\}] \leftarrow (v_L - v_R)/2$ 
18:    end for
19:  end if
20:  return  $\mathcal{N}$ 
21: end procedure
```

478 A.3 Sparse Fourier Optimization

We assume $\hat{f}(S)$ is a sparse, low-degree function with support \mathcal{K} :

$$\hat{f}(S) = \sum_{T \in \mathcal{K}} (-1)^{|S \cap T|} \hat{F}(T)$$

Equivalently, the function can be represented (and efficiently converted) under the Möbius transform. Converting Fourier to Möbius (via Appendix A.1), letting $\mathcal{K}^+ = \{R \subseteq T \mid T \in \mathcal{K}\}$, and applying the inverse Möbius transform:

$$\hat{f}(S) = \sum_{R \in \mathcal{K}^+, R \subseteq S} \hat{I}^M(R)$$

The optimization problem can then be expressed as a polynomial over $\{0,1\}$. Let \mathbf{x} be a binary vector of length n and $S = \{i \in [n] \mid x_i = 1\}$. We will focus on the maximization problem (minimization follows analogously).

$$\max_{S \subseteq [n]} \hat{f}(S) = \max_{\mathbf{x} \in \{0,1\}^n} \sum_{R \in \mathcal{K}^+} \hat{I}^M(R) \prod_{i \in R} x_i$$

479 To reduce the problem to a linear integer program, each monomial $\prod_{i \in R} x_i$ can be replaced with a
480 decision variable y_R and the following constraints:

$$\max_{\mathbf{y} \in \{0,1\}^{|\mathcal{K}^+|}} \sum_{R \in \mathcal{K}^+} \hat{I}^M(R) y_R \tag{7}$$

$$\text{s.t. } y_R \leq y_Q \quad \forall Q \subset R, R \in \mathcal{K}^+ \tag{8}$$

$$\sum_{i \in R} y_{\{i\}} < |R| + y_R \quad \forall R \in \mathcal{K}^+ \tag{9}$$

481 The first constraint guarantees that whenever a monomial is activated (i.e. $x_i = 1 \ \forall i \in R$), all
482 of its subsets are also activated. The second constraint ensures that if a monomial is deactivated
483 (i.e. $\exists i \in R$ s.t. $x_i = 0$), at least one of its constituent terms ($y_{\{i\}}$) is likewise deactivated. After
484 the optimization is solved, the solution can be read-off from the univariate monomials $y_{\{i\}}$. These
485 monomial terms can also be used to impose cardinality constraints on the solution, as was used in
486 Section 4.2 and Section 5.2.

B Experimental Details

B.1 Implementation Details

B.1.1 Hyper-parameters

We performed 5-fold cross-validation over the following hyper-parameters for each of the models:

Model	Hyper-parameter
LASSO	L1 Reg. Param. λ (100 with $\lambda_{min}/\lambda_{max} = 0.001$)
SPEX	L1 Reg. Param. λ (100 with $\lambda_{min}/\lambda_{max} = 0.001$)
PROXYSPEX	Max. Tree Depth [3, 5, None]
	Number of Trees [500, 1000, 5000]
	Learning Rate [0.01, 0.1]
	L1 Reg. Param. λ (100 with $\lambda_{min}/\lambda_{max} = 0.001$)
Random Forest	Max. Tree Depth [3, 5, None]
	Number of Trees [100, 500, 1000, 5000]
Neural Network	Hidden Layer Sizes [$(\frac{n}{4})$, $(\frac{n}{4}, \frac{n}{4})$, $(\frac{n}{4}, \frac{n}{4}, \frac{n}{4})$]
	Learning Rate [Constant, Adaptive]
	Learning Rate Init. [0.001, 0.01, 0.1]
	Number of Trees [100, 500, 1000, 5000]

B.1.2 Sentiment Analysis

20 movie reviews were used from the *Large Movie Review Dataset* [45] with $n \in [256, 512]$ words. To measure the sentiment of each movie review, we utilize a DistilBERT model [46] fine-tuned for sentiment analysis [47]. When masking, we replace the word with the [UNK] token. We construct an value function over the output logit associated with the positive class.

B.1.3 HotpotQA

We consider 50 examples from the *HotpotQA* [48] dataset between $n \in [64, 128]$ sentences. We use a Llama-3.2-3B-Instruct model with 8-bit quantization. When masking, we replace with the [UNK] token, and measure the log-perplexity of generating the original output. Since *HotpotQA* is a multi-document dataset, we use the following prompt format.

Title: {title_1}
Content: {document_1}
 ...
Title: {title_m}
Content: {document_m}

Query: {question}. Keep your answers as short as possible.

B.1.4 DROP

We consider 50 examples from the *DROP* [48] dataset with $n \in [256, 512]$ number of words. We use the same model as *HotpotQA* and mask in a similar fashion. We use the following prompt format.

Context: {context}
Query: {question}. Keep your answers as short as possible.

506 B.1.5 MS-COCO

507 We utilize the Microsoft Common Objects in Context (MS-COCO) dataset [41], which comprises
 508 images paired with descriptive text captions. For our experiments, we treat image patches (there are
 509 48 patches per image) and individual words from the captions as the input features. We used the first
 510 50 examples from the test set, which had n (image patches + words) between the range of [60, 85].

511 To model the relationship between images and text, we employed the CLIP-ViT-B/32 model, a
 512 vision-language encoder designed to learn joint representations of visual and textual data. In our
 513 PROXYSPEX framework, when masking input features (either image patches or words), we replace
 514 them with a generic placeholder token suitable for the CLIP architecture (e.g., a zeroed-out patch
 515 vector or the text [MASK] words. The value function $f(S)$ for a given subset of features S was
 516 defined as the contrastive loss among the other image/caption pairs. By measuring the change in this
 517 contrastive loss upon masking different feature subsets, we can attribute importance to individual
 518 features and their interactions in the context of joint image-text understanding.

519 B.2 Measuring Spectral Hierarchies

520 To quantify the hierarchical structure observed in the Fourier spectra of the LLMs under study,
 521 we introduce and analyze two key metrics: the Staircase Rate (SCR) and the Strong Hierarchy
 522 Rate (SHR). These metrics are computed based on the set of the k largest (in magnitude) Fourier
 523 coefficients, denoted as \mathcal{F}_k .

524 The *Staircase Rate* ($SCR(f, k)$) is defined as:

$$SCR(f, k) = \frac{1}{k} \sum_{S \in \mathcal{F}_k} \mathbb{1} \left\{ \exists (e_1, \dots, e_{|S|}) \in \text{Perm}(S) \text{ s.t. } \left(\forall j \in \{0, \dots, |S|\} : \bigcup_{l=1}^j \{e_l\} \in \mathcal{F}_k \right) \right\},$$

where \mathcal{F}_k denotes the k largest Fourier coefficients of f ,
 and $\text{Perm}(S)$ is the set of all ordered sequences of the elements in S .

(10)

525 The SCR measures the proportion of top- k Fourier coefficients $F(S)$ for which there exists an
 526 ordering of its constituent elements $(e_1, \dots, e_{|S|})$ such that all initial subsets (i.e., $e_1, \{e_1, e_2\}, \dots, S$
 527 itself) are also among the top- k coefficients. A high SCR indicates that significant high-order
 528 interactions are built up from significant lower-order interactions in a step-wise or "staircase" manner.

529 The *Strong Hierarchy Rate* ($SHR(f, k)$) is defined as:

$$SHR(f, k) = \frac{1}{k} \sum_{S \in \mathcal{F}_k} \mathbb{1} \{ \forall S' \subseteq S, S' \in \mathcal{F}_k \}, \quad \text{where } \mathcal{F}_k \text{ denotes the } k \text{ largest Fourier coefficients of } f. \quad (11)$$

530 The SHR is a stricter measure, quantifying the proportion of top- k coefficients $F(S)$ for which all
 531 subsets of S (not just initial subsets, as in DSR) are also present in \mathcal{F}_k . A high SHR suggests a very
 532 robust hierarchical structure where the significance of an interaction implies the significance of all its
 533 underlying components.

534 Figure 10 visualizes these rates alongside faithfulness (R^2) for the Sentiment Analysis and MS-
 535 COCO datasets. These empirical results aim to demonstrate that LLM feature interactions exhibit
 536 significant hierarchical structure. The high SCR and SHR scores support the core motivation for
 537 PROXYSPEX: that important interactions are often built upon their lower-order subsets, a structure
 538 that Gradient Boosted Trees (GBTs) are well-suited to capture and exploit.

539 B.3 Sparsification

540 The process of sparsification is crucial for enhancing the interpretability of the explanations generated
 541 by PROXYSPEX. By retaining only the top k Fourier coefficients, we can achieve a more concise
 542 and understandable representation of the model's behavior without significantly compromising
 543 the faithfulness of the explanation. As demonstrated in Figure 5, a relatively small number of
 544 Fourier coefficients (approximately 200) are often sufficient to achieve faithfulness comparable to
 545 using a much larger set of coefficients for tasks like sentiment classification and image captioning
 546 (MS-COCO).

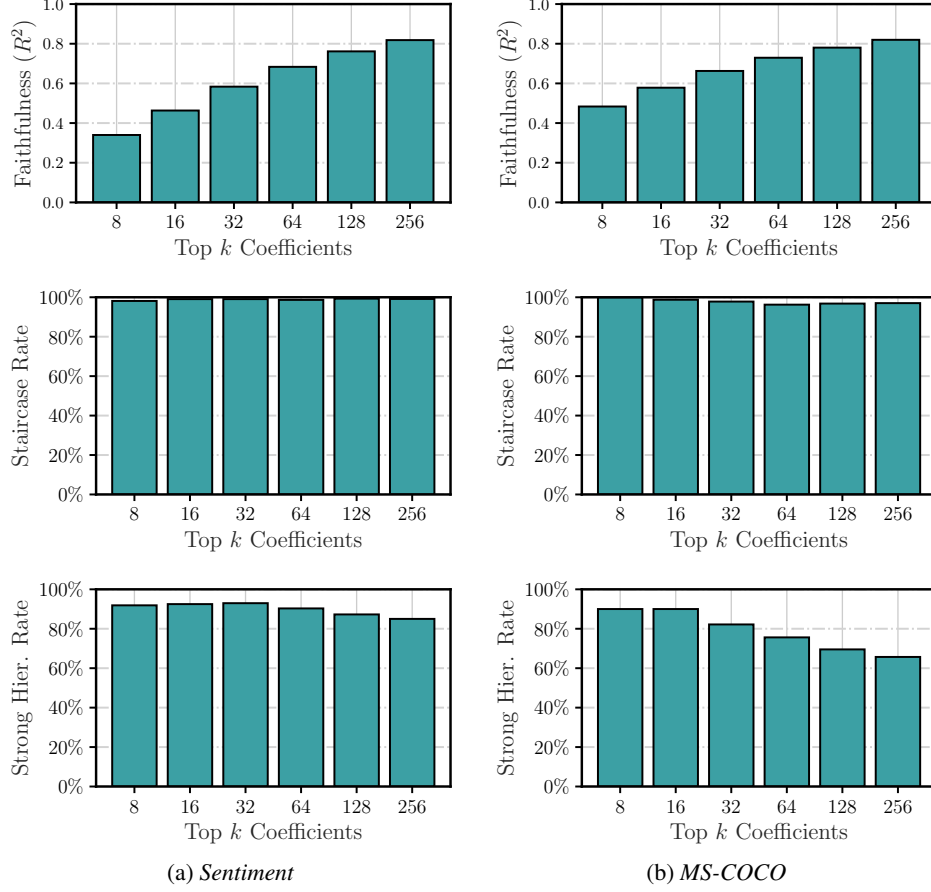


Figure 10: (top row) We run SPEX until $R^2 > 0.9$. We report the faithfulness of when we truncate the spectrum to keep just the top k coefficients for a range of k . We include results from Sentiment $n \in [256, 512]$, and MS-COCO $n \in [60, 85]$. In both cases faithfulness steadily increases as we increase k . (middle row) We report the SCR (10) for the same top k Fourier truncated functions above. In all cases, the SCR is nearly 100%. (bottom row) We also report the SHR (11), which is the strongest of the metric we consider. Here we find that even though SHR decreases somewhat as k grows, it is still strongly in favor of the hierarchy hypothesis.

Further results in Figure 11 illustrate the relationship between relative faithfulness and Fourier sparsity for both Sentiment and MS-COCO datasets across different inference multipliers (α). These plots show that faithfulness generally increases with k , plateauing after a certain number of coefficients, reinforcing the idea that a sparse representation can effectively capture the essential dynamics of the LLM’s decision-making process.

B.4 Proxy Model Selection

The choice of GBTs as the proxy model within PROXYSPEX is motivated by their inherent ability to identify and learn hierarchical interactions from limited training data. This is a critical characteristic, as LLM feature interactions often exhibit a hierarchical structure where higher-order interactions are built upon their lower-order subsets. As indicated in the main text, GBTs have been shown to vastly outperform other proxy models, including random forests, particularly because random forests are less effective at learning hierarchical functions. GBT-like algorithms, on the other hand, are adept at disentangling sums of these hierarchical components.

Figure 12 provides a comparative view of proxy model performance. Figure 12a and Figure 12b illustrate the faithfulness (R^2) of different proxy models (LASSO, Random Forest, Neural Network,

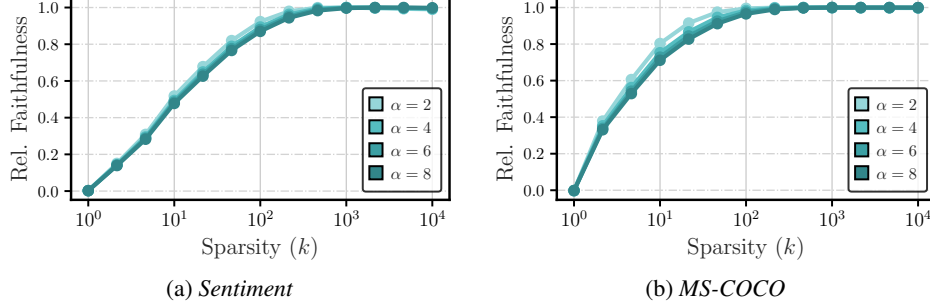


Figure 11: We plot faithfulness (R^2) as a function of Fourier sparsity. Only ≈ 200 coefficients are required to achieve equivalent faithfulness.

and GBTs) on both a synthetic dataset with a complete hierarchy (defined below) and the Sentiment Analysis dataset, respectively, across various inference parameters (α). These results empirically support the superiority of GBTs in capturing these complex interaction structures. However, it’s also important to acknowledge limitations; for instance, GBTs may not perform as well when interactions possess a different, non-hierarchical sparse structure, as empirically confirmed by simulations like the Synthetic-Peak example (which lacks hierarchical structure) shown in Figure I2c.

Synthetic Peak	Synthetic Complete Hierarchy
$f^{\text{SP}}(S) = \sum_{T \subseteq \mathcal{P}} (-1)^{ S \cap T } F(T)$ where \mathcal{P} is a set of 10 uniformly sampled sets of cardinality 5 and $F(T) \sim \text{Uniform}(-1, 1)$ for $T \in \mathcal{P}$	$f^{\text{SCH}}(S) = \sum_{R \subseteq \mathcal{H}} (-1)^{ S \cap R } F(R)$ where $\mathcal{H} = \{R \subseteq T \mid T \in \mathcal{P}\}$ and $F(R) \sim \text{Uniform}(-1, 1)$ for $R \in \mathcal{H}$

B.5 Practical Implications

The practical implications of PROXYSPEX are significant, primarily revolving around its inference efficiency and the resulting speedups in generating faithful explanations for LLMs. A major challenge with existing interaction attribution methods, like SPEX, is the substantial number of model inferences required, which can be computationally expensive and time-consuming for large models. PROXYSPEX addresses this by leveraging a GBT proxy model, which dramatically reduces the number of inferences needed while maintaining or even improving explanation faithfulness.

Figure I3 presents the practical benefits in terms of wall clock time for achieving different levels of faithfulness (R^2) on the Sentiment Analysis (Figure I3a) and MS-COCO (Figure I3b) datasets. These plots clearly demonstrate the speedups achieved by PROXYSPEX. For example, in the sentiment analysis task using the smaller DistilBERT model, PROXYSPEX offers a speedup of approximately 3x, while for the larger CLIP-ViT-B/32 model with MS-COCO, the speedup is around 5x when compared to methods that require more extensive sampling. This increased efficiency makes PROXYSPEX a more viable tool for interpreting complex LLMs in real-world scenarios where computational resources and time are often constrained.

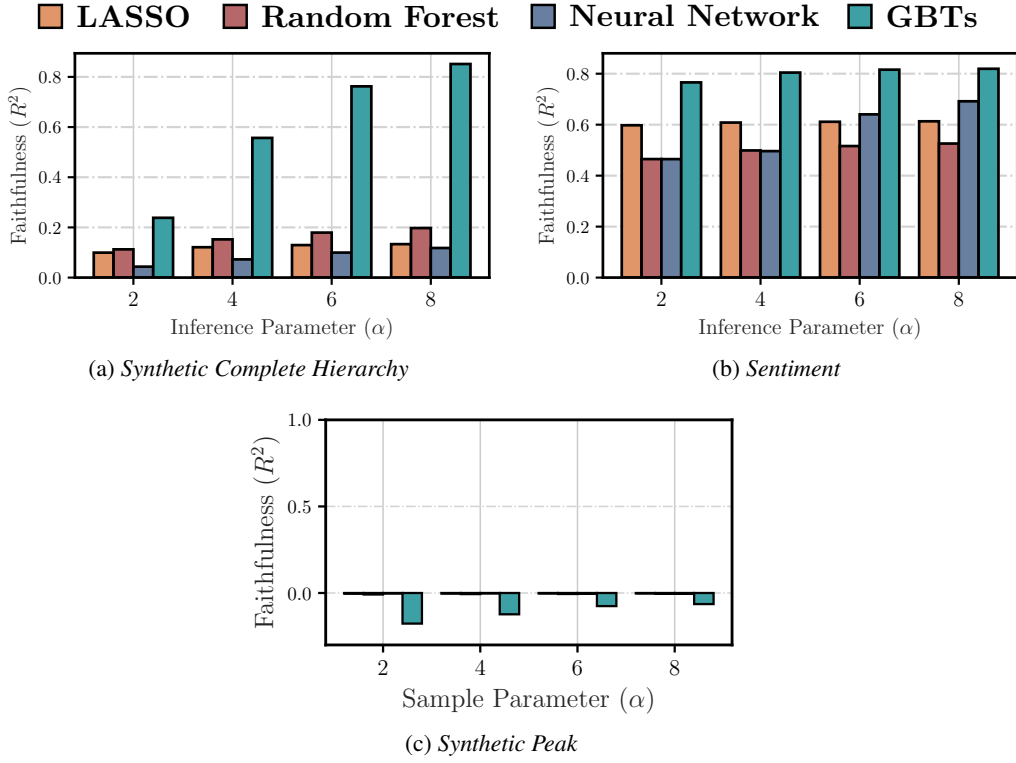


Figure 12: Comparison of proxy model faithfulness in capturing function structures. (a) Faithfulness of LASSO, Random Forest, Neural Network, and GBTs on a synthetic dataset with a complete hierarchical structure, across varying inference parameters (α). (b) Faithfulness of the same proxy models on the Sentiment Analysis dataset across varying α . (c) Faithfulness on a synthetic dataset with a sparse, non-hierarchical peak function, across varying α , illustrating a limitation of GBTs for non-hierarchical structures.

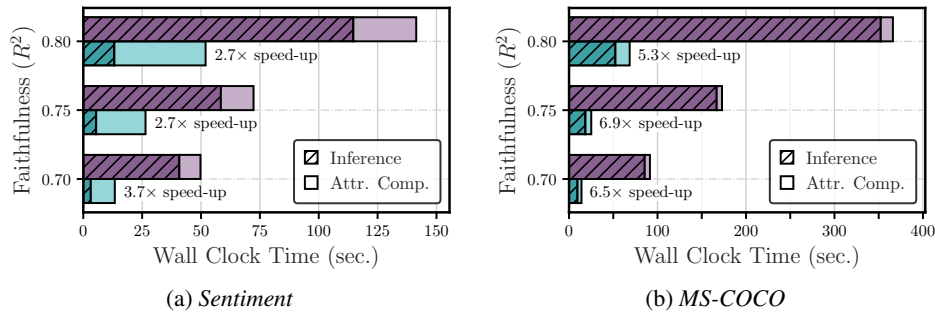


Figure 13: Wall clock time demonstrating PROXYSPEX's efficiency. Comparison of wall clock time (seconds) required to achieve different levels of faithfulness (R^2) for PROXYSPEX, showing breakdown of inference time and attribution computation time. (a) Results on the Sentiment Analysis dataset with the DistilBERT model. (b) Results on the MS-COCO dataset with the CLIP-ViT-B/32 model, highlighting speedups achieved by PROXYSPEX.

C Case Study Details

C.1 Data Attribution via Non-Linear Datamodels

The training masks and margin outputs were provided by [52], corresponding to their subsampling rate of 50% (i.e., half the training images were used to fit each model). See [52] for the hyperparameters selected. With $n = 50,000$ training samples, 300,000 training masks (model retrainings) were provided. This corresponds to $\alpha \approx 0.38$, which underscores the inference-efficiency of PROXYSPEX to identify strong interactions.

Utilizing these masks and margins, we randomly selected 60 test images (6 from each class) for analysis with PROXYSPEX. Below, in Figure 14 and Figure 15 we present the strongest second-order interactions of the first thirty of these selected test images. Figure 8 visualizes the six test images exhibiting the most significant third-order interactions identified through this analysis.

After fitting PROXYSPEX, we convert the Fourier interactions to Möbius using Appendix A.1. Since target and non-target images affect the test margin in opposite directions, we partition the interaction space into the following categories:

- *Target-class interactions* \mathcal{T} : Interactions composed exclusively of training images that share the same label as the held-out test image.
- *Non-target-class interactions* \mathcal{T}^c : Interactions where at least one training image in the set has a label different from that of the held-out test image.

Synergistic Interactions: The top synergistic interaction R^* of order- r is defined as:

$$\begin{aligned} S^* &= \operatorname{argmax}_{S \in \mathcal{T}, |S|=r} I^M(S) \\ T^* &= \operatorname{argmin}_{T \in \mathcal{T}^c, |T|=r} I^M(T) \\ R^* &= \begin{cases} S^* & \text{if } |I^M(S^*)| \geq |I^M(T^*)| \\ T^* & \text{otherwise} \end{cases} \end{aligned} \quad (12)$$

Visually, as presented in Figure 14 for $r = 2$, the interactions R^* identified by this rule often involve training images that appear to work together to reinforce or clarify the classification of the held-out image, frequently by contributing complementary features or attributes. It is important to acknowledge that this definition serves as a heuristic and does not perfectly isolate synergy; For example, the first frog image contains redundant bird images due to strong higher-order interactions involving these bird images.

Redundant Interactions: The top redundant interaction R^* of order- r is defined as:

$$\begin{aligned} S^* &= \operatorname{argmin}_{S \in \mathcal{T}, |S|=r} I^M(S) \\ T^* &= \operatorname{argmax}_{T \in \mathcal{T}^c, |T|=r} I^M(T) \\ R^* &= \begin{cases} S^* & \text{if } |I^M(S^*)| \geq |I^M(T^*)| \\ T^* & \text{otherwise} \end{cases} \end{aligned} \quad (13)$$

Figure 15 demonstrates that this definition identifies redundant training images that are similar to the held-out image.

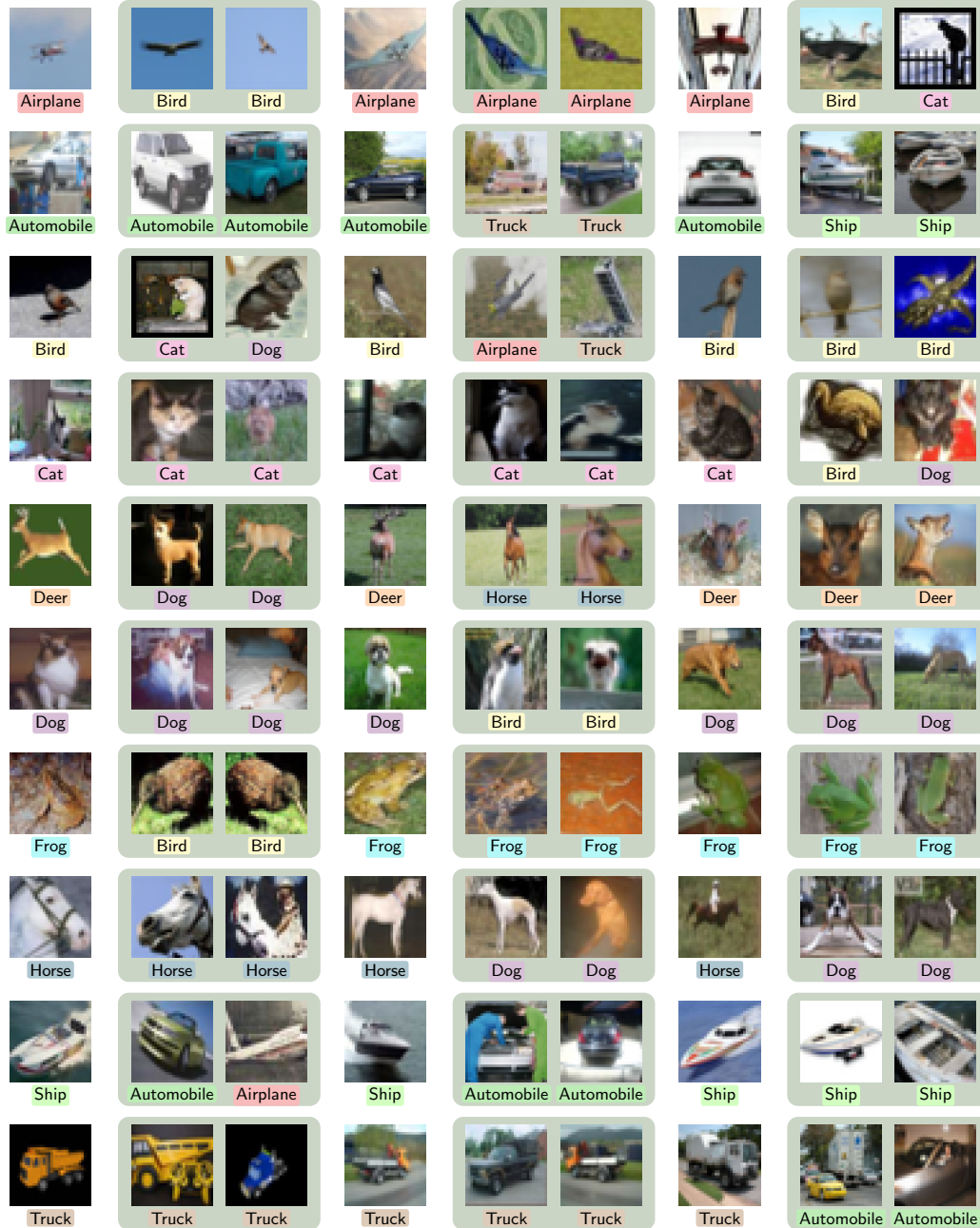


Figure 14: For 30 random held-out images, their corresponding top second-order *synergistic interaction* (green box).

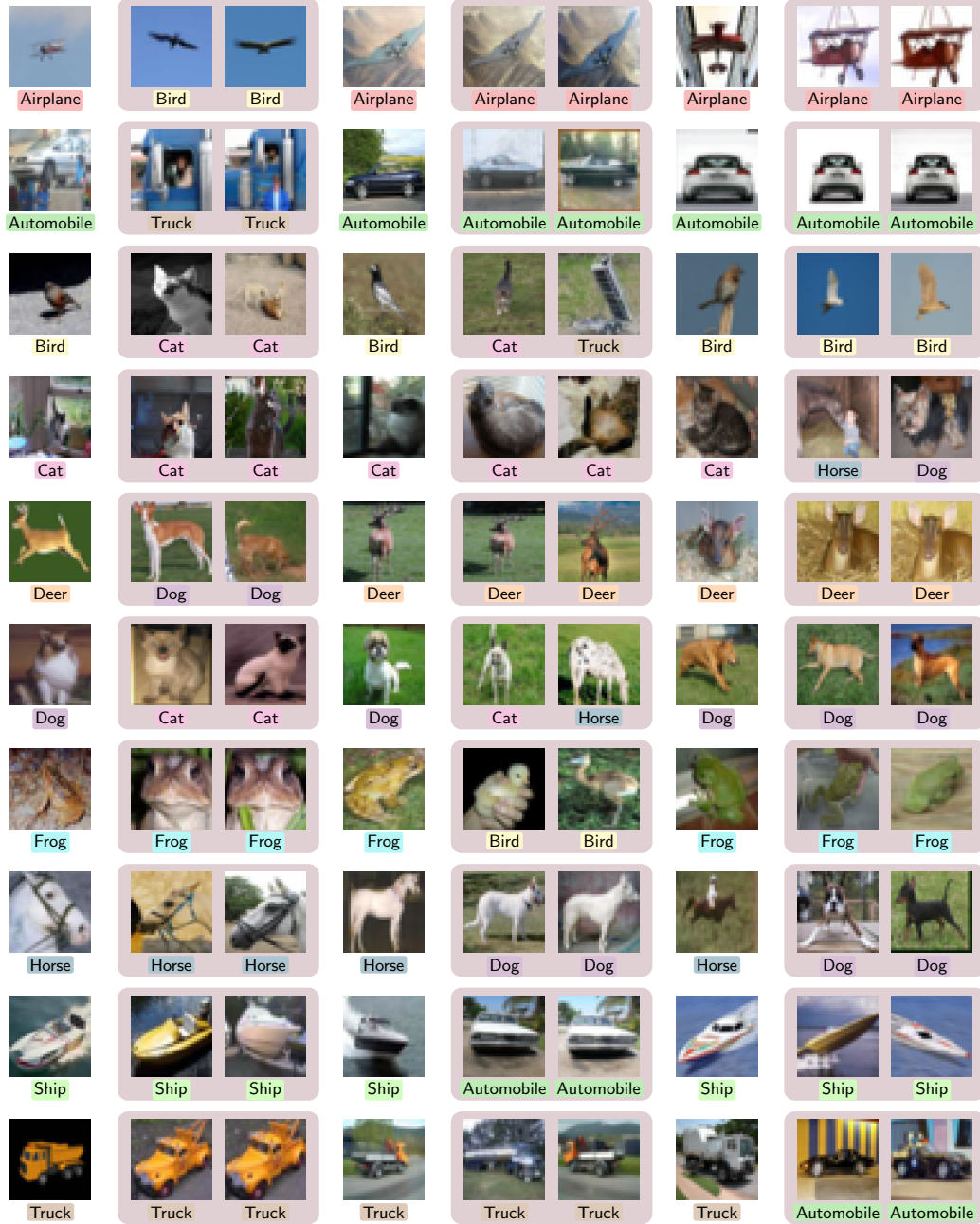


Figure 15: For 30 random held-out images, their corresponding top second-order *redundant interaction* (red box).

C.2 Model Component Attribution

We study the influence of specific model components on task performance, using a controlled ablation methodology. Our experiments are conducted on Llama-3.1-8B-Instruct evaluated on the high-school-us-history subset of the MMLU dataset, a benchmark comprising multiple-choice questions.

MMLU includes 231 questions in the high-school-us-history subset. To perform pruning and then evaluate the ablated models, we split this data into two sets—training split $\mathcal{D}_{\text{train}}$ consisting of the first 120 questions and test split $\mathcal{D}_{\text{test}}$ with the remaining questions. We use accuracy as the evaluation metric, which is computed as the proportion of correctly answered multiple-choice questions on a given data split.

For an L layer LLM, we let $[L]$ denote the set of layers and let \mathcal{H}_ℓ denote the set of attention heads in layer $\ell \in [L]$. For each experiment, we focus on a particular group of layers $\mathcal{L} \subseteq [L]$ within the model and denote the corresponding set of attention heads as $\mathcal{H}_\mathcal{L} = \bigcup_{\ell \in \mathcal{L}} \mathcal{H}_\ell$. The Llama-3.1-8B-Instruct model consists of $L = 32$ layers, each with 32 attention heads.

At each layer ℓ of the LLM, the output of the attention heads is combined into a latent representation by concatenating the outputs of the attention heads. Then, this latent vector is passed to the feed-forward network of layer ℓ . To study the contribution of specific heads, we define an ablated model LLM_S for any subset $S \subseteq \mathcal{H}_\mathcal{L}$. In LLM_S , the outputs of the heads in $\mathcal{H}_\mathcal{L} \setminus S$ are set to zero before the concatenation step. After concatenation, we apply a rescaling factor to the resulting latent vector at each layer $\ell \in \mathcal{L}$, equal to the inverse of the proportion of retained heads in that layer, i.e., $\frac{|\mathcal{H}_\ell|}{|S \cap \mathcal{H}_\ell|}$. This modified latent representation is then passed to the feed-forward network as usual.

We define $f_\mathcal{L}$ as

$$f_\mathcal{L}(S) \triangleq \text{Accuracy of } \text{LLM}_S \text{ on } \mathcal{D}_{\text{train}}, \quad (14)$$

and interpret $f_\mathcal{L}(S)$ as a proxy for the functional contribution of head subset S to model performance, enabling quantitative analyses of attribution and interaction effects among attention heads.

Pruning. We perform pruning experiments across three different layer groups \mathcal{L} : initial layers ($\mathcal{L} = \{1, 2, 3\}$), middle layers ($\mathcal{L} = \{14, 15, 16\}$), and final layers ($\mathcal{L} = \{30, 31, 32\}$). Since each layer has 32 attention heads, we effectively perform ablation over $n = |\mathcal{H}_\mathcal{L}| = 96$ features (attention heads) in total. For a given group \mathcal{L} , we begin by estimating the function $f_\mathcal{L}$ using both LASSO and PROXSPEX, based on evaluations of $f_\mathcal{L}(S)$ for 5000 subsets S sampled uniformly at random. These estimates serve as surrogates for the true head importance function. We then maximize the estimated functions to identify the most important attention heads under varying sparsity constraints (target numbers of retained heads). We use the procedure detailed in Section 4.2 to identify heads to remove for both PROXSPEX and LASSO. We also compare against a Best-of- N baseline, in which the model is pruned by selecting the subset S that achieves the highest value of $f_\mathcal{L}(S)$ among 5000 randomly sampled subsets at the target sparsity level.

Evaluation. In order to evaluate the performance of an ablated model LLM_S , we measure its accuracy on the test set using

$$g_\mathcal{L}(S) \triangleq \text{Accuracy of } \text{LLM}_S \text{ on } \mathcal{D}_{\text{test}}. \quad (15)$$

In Figure 9, we report the value of $g_\mathcal{L}(S)$ for the pruned models obtained by each method. We find that PROXSPEX consistently outperforms both baselines, yielding higher test accuracy across all evaluated sparsity levels.

Inference setup. All experiments are run on a single NVIDIA H100 GPU, with batch size 50. Average runtime per ablation (i.e., evaluating $f_\mathcal{L}(S)$ once for a given S) is approximately 1.7 seconds. Therefore, collecting a training dataset $\{(S_i, f_\mathcal{L}(S_i))\}$ with 5000 training samples takes approximately 2.5 hours.