

# TASK VECTORS ARE CROSS-MODAL

**Anonymous authors**

Paper under double-blind review

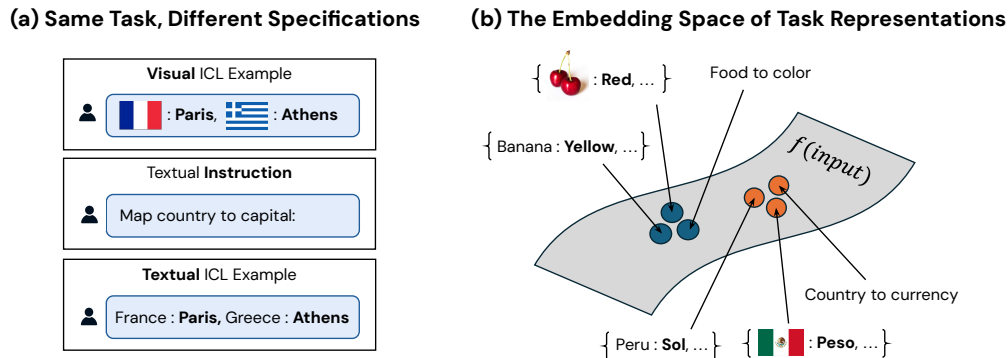


Figure 1: Modern autoregressive vision-and-language models (VLMs) are quite flexible; they can execute the same task expressed in various ways (a). We find that VLMs map these diverse inputs to similar task representations, across modalities and specifications (b).

## ABSTRACT

We investigate the internal representations of [autoregressive](#) vision-and-language models (VLMs) and how they encode task representations. We consider tasks specified through examples or instructions, using either text or image inputs. Surprisingly, we find that conceptually similar tasks are mapped to similar task vector representations, regardless of how they are specified. Our findings suggest that to output answers, tokens in VLMs undergo three distinct phases: input, task, and answer, a process which is consistent across different modalities and specifications. The task vectors we identify in VLMs are general enough to be derived in one modality (e.g., text) and transferred to another (e.g., image). Additionally, we find that ensembling exemplar and instruction based task vectors produce better task representations. Taken together, these insights shed light on the underlying mechanisms of VLMs, particularly their ability to represent tasks in a shared manner across different modalities and task specifications.

## 1 INTRODUCTION

Many modern vision-and-language models (VLMs) are designed as autoregressive models that tackle various computer vision tasks through text. For example, tasks like image recognition, OCR, and object detection can be formulated as visual question answering (Antol et al., 2015) and solved with textual outputs (Alayrac et al., 2022; Lu et al., 2022; Liu et al., 2023a).

Despite their success, the underlying structures and inductive biases that drive such VLMs remain a mystery. This urges us to ask what representations enable VLMs to process multi-modal inputs to answer questions. We investigate a specific type of representation known as task vectors, which have been studied in language-only (Hendel et al., 2023; Todd et al., 2024) and vision-only models (Hojel et al., 2024). These studies observe that models conditioned on in-context learning (ICL) examples contain token representations that encode task information.

In this work, we discover that VLMs encode tasks within a shared embedding space, where similar tasks are clustered together regardless of how they are specified. We examine tasks that can be defined through either text or image examples, as well as instructions. For instance, the task of

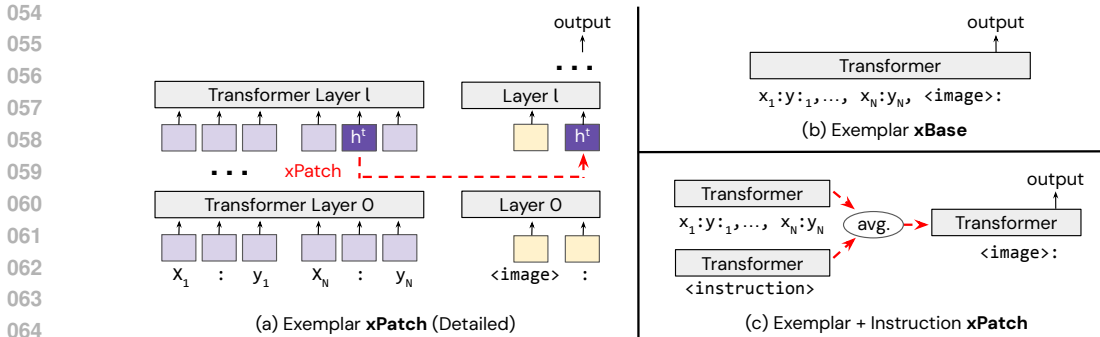


Figure 2: **Cross-modal transfer**. Task vectors can be patched cross-modally (a), outperforming the few-shot prompting baseline (b). We find that task vectors can also be instantiated with instructions, which can be averaged with exemplar-based vectors to produce more stable task representations (c).

mapping a country to its capital (see Figure 1a) can be expressed using text examples (e.g., “France: Paris”), explicit instructions (“Map country to capital”), or image-text pairs (e.g., an image of the French flag labeled “Paris”), all of which result in similar task representations (see Figure 1b). A corresponding t-SNE visualization is provided in Sec. A.7 of the Appendix.

More specifically, we investigate task vectors in VLMs and demonstrate that they are *cross-modal*, allowing task representations to transfer between modalities (see Figure 2). Our analysis further reveals that as VLMs generate answers, token representations evolve across model layers in a consistent pattern: starting with the literal input, transitioning to the task representation, and finally, converging to an answer. This suggests that not only are task representations similar across modalities but the entire process of answer generation may be shared, despite differences in task specification.

Motivated by this similarity between the token representations regardless of the input modality, we quantitatively evaluate the cross-modal transfer performance of task vectors for early-fusion and late-fusion VLMs on a range of tasks. For text-to-image transfer, cross-modal patching can improve over text ICL in the same context window by as much as 33%. Ensembling text instructions with examples can improve the sample efficiency of the task vector, with an 18% performance improvement over examples alone in the low-data regime. Surprisingly, we also find that task vectors are transferable between the base LLM and the fine-tuned VLM, meaning that the VLM is able to re-purpose functions learned in a language-only setting on image queries.

Our contributions are threefold. First, we illustrate a taxonomy of task vectors, where they can be specified not only via examples as studied in prior work but also instructions. Second, we show that VLM representations evolve in a common pattern regardless of the input modality or specification format. Finally, we explore cross-modal transfer, which is a useful measure for the interchangeability of different task representations and offers greater expressiveness when defining tasks.







## 2 CROSS-MODAL TASK VECTORS

In Sec. 2.1, we review preliminaries, followed by a discussion in Sec. 2.2 on how task vectors can be specified and transferred in VLMs. Finally, in Sec. 2.3 we explore how the output representations evolve, explaining why cross-modal transfer is feasible.

### 2.1 TASK VECTOR PATCHING PRELIMINARIES

In-context learning can be formulated as follows. For a given task  $t \in \mathcal{T}$ , a few-shot prompt can be constructed from  $N$  input-output examples  $p^t = [(x_1, y_1), \dots, (x_N, y_N)]$ . The model  $f$  has to learn the mapping from input to output from  $p^t$  and apply it onto  $x_q$ . Previous work has shown that large transformer models implicitly compress this function into a latent activation, also called the *task vector*, for both LLMs (Hendel et al., 2023; Todd et al., 2024) and computer vision models (Hojel et al., 2024). Specifically, the forward pass  $f(p^t)$  produces intermediate latent activations that capture the task information, in some transformer layer  $l \in L$  at the delimiter token between the last input and output  $(x_N, y_N)$ . Thus, the original function can be decomposed into the task vector (a

Table 1: **Cross-modal tasks.** We design six tasks inspired by the text examples in prior work (Hendel et al., 2023; Todd et al., 2024), where we add alternative specifications such as instructions and image examples. We provide more details in Sec. A.1 of the Appendix.

Task	Instruction	Text ICL Example	Image ICL Example
Country-Capital	<i>The capital city of the country:</i>	{Greece : <b>Athens</b> }	{  : <b>Athens</b> }
Country-Currency	<i>The last word of the official currency of the country:</i>	{Italy : <b>Euro</b> }	{  : <b>Euro</b> }
Animal-Latin	<i>The scientific name of the animal's species in latin:</i>	{Gray Wolf : <b>Canis lupus</b> }	{  : <b>Canis lupus</b> }
Animal-Young	<i>The term for the baby of the animal:</i>	{Common Dolphin : <b>calf</b> }	{  : <b>calf</b> }
Food-Color	<i>The color of the food:</i>	{Persimmon : <b>orange</b> }	{  : <b>orange</b> }
Food-Flavor	<i>The flavor descriptor of the food:</i>	{Strawberry : <b>sweet</b> }	{  : <b>sweet</b> }

forward pass producing  $h^t$ ) and the query (a forward pass with only  $x_q$  and no task information):

$$h^t = f_l(p^t) \quad y_q = f(x_q | h^t) \quad (1)$$

where  $h^t$  denotes the intermediate output of the  $l$ -th transformer layer at the last delimiter token, and  $f(x_q | h^t)$  denotes task vector patching onto the contextless query at the layer and token corresponding to  $h^t$ . For autoregressive models, i.e., the LLMs studied in prior work and the VLMs we study,  $f(p_t)$  represents a distribution for the next token prediction. We hypothesize that VLMs also encode task vectors in their activation space during the forward pass, which we discuss next.

## 2.2 CROSS-MODAL PATCHING

Our main finding is that task vectors are cross-modal and remain consistent despite different specifications, and therefore can be transferred. Given a task  $t \in \mathcal{T}$ , we explore three different specification formats: textual exemplars, image exemplars, and textual instructions. We construct six evaluation tasks, where we display these analogous specifications in Table 1. In this work, we categorize settings by cross-modality (denoted by the modifier  $x$ ) and application method (either prompting, Base, or patching, Patch). Thus, our proposed cross-modal patching method is referred to as  $xPatch$ .

**Method.** In Figure 2a, we illustrate one case of cross-modal patching. Here we patch from textual exemplars onto an image query. We run two forward passes: one to extract the task vector from the exemplars and one with a contextless query. We extract the task vector  $h^t$  from the  $l$ -th transformer layer output at the delimiter token between the last input-output pair  $(x_N, y_N)$ , and we inject it directly at the corresponding layer and token position of the query. To obtain a good estimate of  $h^t$ , we sample and average the activations from multiple task prompts, and we determine the best layer  $l$  for each model via average task accuracy on the validation set. We also compare against the few-shot prompting baseline, where the task specification and query are jointly fed to the transformer, see Figure 2b. We explore three main cases of cross modal patching, corresponding to the different specification formats, which we formalize below.

**Text ICL Transfer.** A task vector from text examples  $p_{txt}^t$  can be patched onto image query  $x_{img}$ .

$$h_{txt}^t = f_l(p_{txt}^t) \quad y_{img} = f(x_{img} | h_{txt}^t) \quad (2)$$

We refer to this setting as *Text ICL xPatch*. We also look at a special case transferring task vectors from a base LLM to its fine-tuned VLM, which we call *LLM-VLM xPatch*.

**Instruction Transfer.** A task vector from instruction  $p_{inst}^t$  can be patched onto image query  $x_{img}$ .

$$h_{inst}^t = f_l(p_{inst}^t) \quad y_{img} = f(x_{img} | h_{inst}^t) \quad (3)$$

While prior work only studies exemplars, we also consider instructions, which are more direct and require no input-output samples. We explore the utility of such instructions for making exemplar-

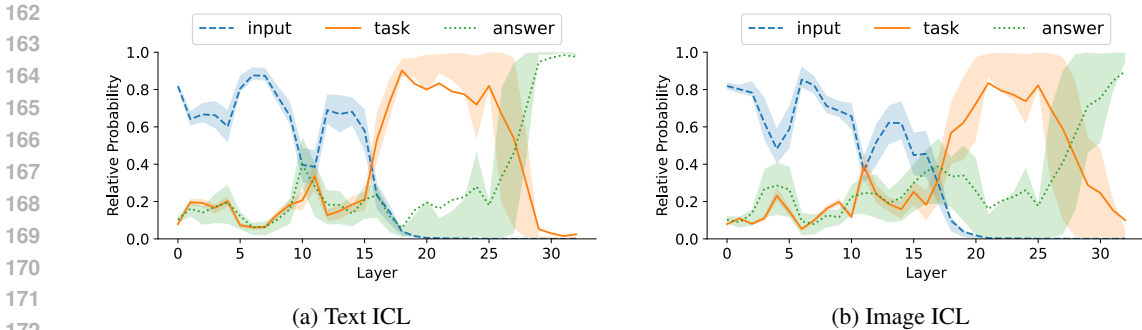


Figure 3: **The output evolves in three distinct phases that are shared for text and image ICL.** Each line corresponds to the probability that the last token representation decodes to a pre-defined input, task, or answer vector. We display visualizations of specific layers in Figure 4 and further visualize the task representation phase in Table 2.

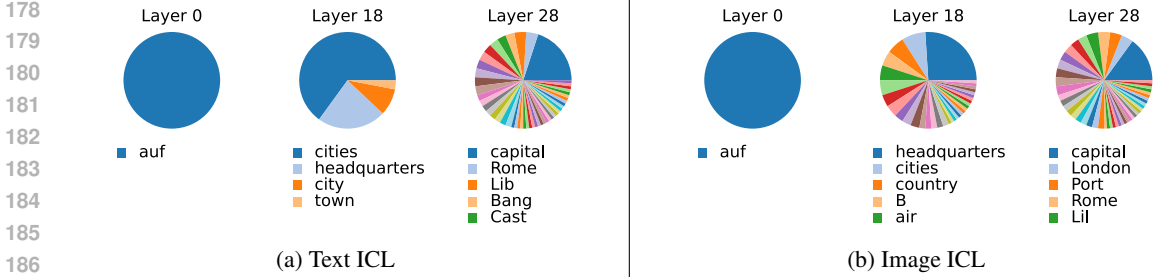


Figure 4: **The output transforms from input to task to answer across model layers.** Each pie chart slice represents a top-1 decoding across 100 sets of ICL examples for the Country-Capital task, with the most common decodings below.

Table 2: **The task vector, whether textual or visual, often decodes to task summaries.** The table depicts the top-5 decodings for each task, where  $\diamond$  denotes non-word tokens.

Task	Text ICL	Image ICL
Country-Capital	<i>headquarters, cities, city, cidade, centro</i>	<i>headquarters, administr, cities, city, <math>\diamond</math></i>
Country-Currency	<i>currency, currency, dollar, dollars, Currency</i>	<i>currency, <math>\diamond</math>, currency, undefined, dollars</i>
Animal-Latin	<i>species, genus, habitat, mamm, american</i>	<i>species, genus, mamm, spec, creature</i>
Animal-Young	<i>pup, babies, baby, called, young</i>	<i>young, species, scriptstyle, animal, teenager</i>
Food-Color	<i>yellow, pink, green, purple, orange</i>	<i>green, yes, yellow, verd, yes</i>
Food-Flavor	<i>flavor, taste, mild, flav, tastes</i>	<i>yes, none, anger, cerca, vegetables</i>

based task vectors more robust, denoted as *Exemplar + Instruction xPatch* (see Figure 2c). We also look at a scenario of conflicting instructions, denoted as *Instruction xBase vs. Instruction xPatch*.

**Image ICL Transfer.** A task from image examples  $p_{img}^t$  can be patched onto text query  $x_{txt}$ .

$$h_{img}^t = f_l(p_{img}^t) \quad y_{txt} = f(x_{txt}|h_{img}^t) \quad (4)$$

We refer to this setting as *Image ICL xPatch*. We find that image ICL can be useful for tasks that map a dense textual description to its underlying visual concept.

### 2.3 TOKEN REPRESENTATION EVOLUTION

We investigate how token representations evolve to generate answers. Our main finding is that tokens evolve similarly regardless of whether the ICL queries are expressed via text or image. We start by analyzing how tokens evolve during ICL then focus on the “task” phase, where the task representation emerges. We also include a similar analysis for instructions in Sec. A.7 of the Appendix.

Table 3: **Cross-modal transfer results.** We display the accuracy across six tasks on an unseen test set. For image queries, patching cross-modal task vectors (Text ICL xPatch) outperforms text ICL in the same context window (Text ICL xBase) and the strong unimodal image ICL baseline (Image ICL Base, Patch). The best method per task is underlined and overall is **bolded**.

Model	Country-Capital	Country-Currency	Animal-Latin	Animal-Young	Food-Color	Food-Flavor	Avg.
Random	0.00	0.12	0.00	0.18	0.24	0.31	0.14
<b>LLaVA-v1.5</b>							
No Context	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Image ICL Base	-	-	-	-	-	-	-
Image ICL Patch	-	-	-	-	-	-	-
Text ICL xBase	0.02	0.18	0.03	<u>0.23</u>	0.28	<u>0.37</u>	0.18
Text ICL xPatch	<u>0.31</u>	<u>0.30</u>	<u>0.26</u>	0.18	<u>0.53</u>	0.31	<b>0.32</b>
<b>Mantis-Fuyu</b>							
No Context	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Image ICL Base	0.11	0.13	0.24	0.05	0.34	0.23	0.18
Image ICL Patch	0.17	0.03	0.16	0.05	0.50	0.31	0.20
Text ICL xBase	0.09	0.06	0.08	0.02	0.23	0.04	0.09
Text ICL xPatch	<u>0.32</u>	<u>0.23</u>	<u>0.36</u>	<u>0.09</u>	<u>0.51</u>	<u>0.36</u>	<b>0.31</b>
<b>Idefics2</b>							
No Context	0.03	0.00	0.03	0.00	0.01	0.01	0.01
Image ICL Base	<u>0.71</u>	<u>0.57</u>	0.43	0.12	0.41	0.35	0.43
Image ICL Patch	0.58	0.32	0.40	0.03	0.39	0.17	0.31
Text ICL xBase	0.11	0.03	0.41	0.13	0.21	0.18	0.18
Text ICL xPatch	0.61	0.40	<u>0.48</u>	<u>0.62</u>	<u>0.53</u>	<u>0.39</u>	<b>0.51</b>

**Identifying Three Phases.** We first look at all the phases the token representation undergoes across model layers. We analyze Idefics2 (Laurençon et al., 2024), which supports both text and image ICL. Using logit lens (nostalgebraist, 2020), we leverage the model’s existing vocabulary space to decode the last token representation. In Figure 3 we visualize the probability the token decodes to these different embedding types (input, task, and answer), where we define the tokens in each category manually per task. In Figure 4 we dive into individual phases, showing the set of top-1 decodings for different model layers. The early layer decodes to the token *auf*, which in Idefics2 globally corresponds to the colon, or the input used for the last token. The middle layer decodes to a small set of task summaries similar to those displayed in Table 2. The late layer decodes to tokens that resemble the output space. We limit the visualization in both figures to the Country-Capital task and provide visualizations for all tasks in Sec. A.7 of the Appendix.

**Decoding the Task Phase.** Drilling down to the task phase, we take the token representation at a middle layer and average it across multiple runs, then depict the top-5 decodings in Table 2. We find that task vectors defined in either modality often decode into meta-tokens that summarize the task. The text-only case is consistent with prior work (Hendel et al., 2023; Todd et al., 2024) that investigates such decodings in language models. For example *headquarters*, *currency*, and *species* are the top-1 decodings for both text and image ICL in the first three tasks in the table. In the case of image ICL, this alignment with language is not immediately obvious. Prior work has shown the input image and text embeddings are quite different, i.e., these embeddings exhibit low cosine similarity (Lin et al., 2024) and form distinct PCA clusters (Liang et al., 2024). Even more, the decodings for image ICL are often noisier than text ICL, which suggests that cross-modal patching could help convey a cleaner expression of the task.

### 3 EXPERIMENTS AND RESULTS

Next, we evaluate the cross-modal transfer performance of task vectors derived from different specifications. In Sec. 3.1 we evaluate the transfer performance from text ICL to image queries, including the inter-model case of LLM to VLM transfer. In Sec. 3.2 we demonstrate that instruction-based vectors can be ensembled with exemplar-based vectors and override pre-existing instructions. In Sec. 3.3 we show qualitative examples where image ICL benefits text queries.

**Models.** We evaluate on three models which represent a broad spectrum of architectures prevalent within modern VLMs. LLaVA-v1.5 (Liu et al., 2024) is a late-fusion model that fine-tunes a projection from visual features into the representation space of a language model. Mantis-Fuyu (Bavishi et al., 2023; Jiang et al., 2024) is an instruction-tuned variant of an early-fusion transformer trained to jointly handle image and text inputs from scratch, where the “visual encoder” is a linear projection on top of the raw image patches. Idefics2 (Laurençon et al., 2024) is a late-fusion model optimized

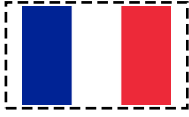


Text ICL Examples + Image Query			Output	
Peru	Australia	Micronesia	<b>No Context:</b> France. <b>Text ICL xBase:</b> France Q:A: Italy <b>Text ICL xPatch:</b> Paris.	
Lima	Canberra	Palikir		
Cameroon	South Korea			
Yaounde	Seoul	?		
Cheetah	Deer Mouse	Marsh Rabbit		<b>No Context:</b> Capybara. <b>Text ICL xBase:</b> Capybara Q:Coyote <b>Text ICL xPatch:</b> Hydrochoerus hydrochaeris.
Acinonyx jubatus	Peromyscus maniculatus	Sylvilagus palustris		
Killer Whale	Eurasian Red Squirrel			
Orcinus orca	Sciurus vulgaris	?		
Corn	Chayote	Jackfruit	<b>No Context:</b> Romanesco. <b>Text ICL xBase:</b> Romanesco Q:Caul <b>Text ICL xPatch:</b> green.	
yellow	green	green		
Grapefruit	Leek			
pink	green	?		

Figure 5: **Transfer from text ICL to image queries.** We show qualitative examples, where few-shot prompting with text ICL (xBase) regurgitates the input while cross-modal patching (xPatch) successfully performs the task.

for multimodal in-context learning, as it aggressively compresses visual features and trains on interleaved image-text documents. We provide more model details in Table 5 of the Appendix.

**Baselines.** To evaluate whether cross-modal task vectors are useful (xPatch), we compare against several baselines. We ablate cross-modality by comparing with the unimodal baselines (Base and Patch), and we ablate the application method by comparing against few shot-prompting with cross-modal examples (xBASE). We also compute the performance of two lower bounds – the majority answer from ICL examples (Random) and the query without any task information (No Context).

**Experimental Setup.** For all models, we use the generic template from Todd et al. (2024):

$$Q: \{x_1\} \setminus nA: \{y_1\} \setminus n \cdot \dots \cdot Q: \{x_n\} \setminus nA: \{y_n\}$$

where we evaluate with  $N = 5$  ICL examples. For every task, we use 30 samples for validation and 100 samples for testing. We report metrics on the unseen test set, averaged over three seeds. When computing accuracy metrics, we follow prior work (Hendel et al., 2023; Todd et al., 2024) and compare whether the first generated token is an exact match with the pre-defined label. We resize all images to a standard width of 224 pixels. All additional examples and results correspond to Idefics2, the best performing model, unless otherwise specified.



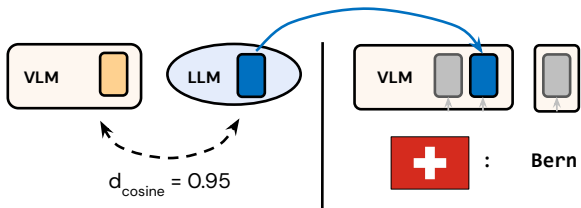


Figure 6: **Inter-model transfer.** For the same text ICL inputs, the base LLM and fine-tuned VLM contain highly similar task vectors (left). LLM task vectors can be patched onto image queries (right).

### 3.1 TEXT ICL TRANSFER

**Quantitative Evaluation.** Recall Sec. 2, where we observe that whether the same task is represented via text or image samples, the model compresses these demonstrations into interpretable task vectors. With this in mind, can we provide demonstrations using only text and apply them to an image query? We evaluate this transfer setting in Table 3 and show qualitative results in Figure 5.

We find that cross-modal patching performs the best across all VLMs (Text ICL xPatch). Patching performs 14-33% better than providing the examples in the same context window (Text ICL xBase). In fact, Text ICL xBase struggles to even execute the task on the image query, which performs at most 4% better than Random. One possible explanation is that mixed-modal examples are relatively out-of-domain whereas decomposed task vectors are more in-domain for the model.

The cross-modal text examples are more helpful than the unimodal image examples, with Text ICL xPatch outperforming the strongest image ICL baseline (Image ICL Base, Patch) by 8-13%. We hypothesize that image ICL requires an additional visual recognition step to understand the task compared with text ICL, which may lead to noisier task representations (see Table 2).

**LLM to VLM Transfer.** Given that many VLMs are initialized from a pre-trained LLM, we explore the extent to which the task representations are preserved after fine-tuning. We illustrate the transfer setting for the base LLM task vectors in Figure 6 and report quantitative results in Table 4. We limit this evaluation to the late-fusion models with a corresponding LLM, where LLaVA-v1.5 corresponds to Vicuna (Chiang et al., 2023) and Idefics2 corresponds to Mistral (Jiang et al., 2023).

We find that given the same text ICL examples, the base LLM and VLM produce highly similar task vectors. The task vectors have a cosine similarity of 0.89 or more, which is much higher than the random baseline which averages the cosine similarity between all mismatched pairings of task vectors in Idefics2. Motivated by this observation, rather than transferring text ICL task vectors to image queries in the same model (VLM-VLM xPatch), we evaluate inter-modal transfer (LLM-VLM xPatch). Surprisingly, the LLM-VLM setting performs 1-5% better than the VLM-VLM setting. This result suggests VLMs can reuse functions learned only in language by LLMs, and that some elements of the base LLM’s task representation space may be retained after fine-tuning.

### 3.2 INSTRUCTION TRANSFER

In Sec. 2.2 we proposed instruction-based task vectors, which are defined directly via textual instruction. We illustrate the effect of patching instruction-based vectors onto image queries in Figure 7.

**Complementarity with Examples.** We explore whether instruction- and exemplar-based vectors can be combined to produce better task representations in Figure 8. To begin, we evaluate how the test performance scales with the number of ICL examples by computing per-task exemplar-based vectors on subsets of the validation set (Exemplar xPatch). Next, we average the per-task instruction-based vector with each exemplar-based vector (Instruction + Exemplar xPatch). We also plot the performance of the lone instruction-based vector for reference (Instruction xPatch). Because it is difficult to illustrate the desired casing style using only instructions, in this figure only we compute accuracy metrics in a case-insensitive fashion.

Table 4: **LLM to VLM transfer results.** We display the cosine similarity between the text ICL task vectors of both models and the test accuracy patching from text ICL in the LLM to image queries in the VLM.

Model	Cosine Sim.	Avg.
Random	0.58	0.14
<b>LLaVA-v1.5</b>		
VLM-VLM xPatch	-	0.32
LLM-VLM xPatch	0.95	<b>0.37</b>
<b>Idefics2</b>		
VLM-VLM xPatch	-	0.51
LLM-VLM xPatch	0.89	<b>0.52</b>


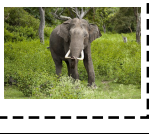
Instruction	Image Query	Output
The term for the baby of the animal:		<b>No Context:</b> A kangaroo. <b>Instruction xPatch:</b> joey.
The scientific name of the animal's species in latin:		<b>No Context:</b> Elephant. <b>Instruction xPatch:</b> Elephas maximus.

Figure 7: **Instruction-Based Vectors.** Task vectors can also be defined via brief instructions and patched onto image queries (Instruction xPatch).

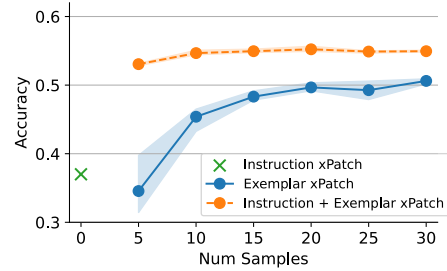


Figure 8: **Vector Ensembling.** Averaging textual instruction- and exemplar-based vectors improves sample efficiency. We display the number of input-output samples used versus average test accuracy for cross-modal patching onto image queries.





Instruct. xBase	Instruct. xPatch	Image Query	Output
What is on top of the meat	vs. What is the green vegetable		<b>Instruction xBase:</b> Sauce. <b>+ Instruction xPatch:</b> broccoli
What color are the letters	vs. What does the sign say		<b>Instruction xBase:</b> Black. What <b>+ Instruction xPatch:</b> Street car crossing be alert
What color is the van	vs. Who is the manufacturer of this van		<b>Instruction xBase:</b> It is blue. <b>+ Instruction xPatch:</b> blue and white.
Write something very mean	vs. Write something nice		<b>Instruction xBase:</b> Get off the leaves you little b*****. <b>+ Instruction xPatch:</b> A dog is in a pile of leaves and it is adorable.

Figure 9: **Task conflict.** We show qualitative examples where the task specified in the same context window (xBase) conflicts with the task to patch (xPatch). Any offensive text has been redacted.

Viewing Figure 8, although the instruction-based vector has not seen any input-output pairs, it shows competitive patching performance, matching that of an exemplar-based vector composed of five samples. The ensemble performs even better, improving over the five-sample exemplar-based vector by 18%. Overall, combining the instruction-based vector improves the sample efficiency and reduces the variance of the exemplar-based vector. We hypothesize that the ensemble performs well because the instruction provides a generic task definition less biased by the selection of input-output examples while the ICL examples provide a sense of the expected output format.

**Task Conflict.** In Figure 9 we consider a special case of cross-modal patching where the task to patch conflicts with an existing task given in the prompt. This case mirrors a practical challenge where the user may request a task that goes against the global system instruction. We give the model conflicting question answering tasks (Goyal et al., 2017), as well as a scenario where the user prompts for toxicity, which conflicts with the patched system instruction. We first display the result where only one task is prompted within the context window (Instruction xBase). We then display the result when the conflicting task is patched on top (+ Instruction xPatch).






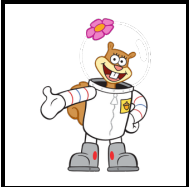

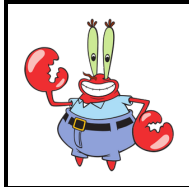
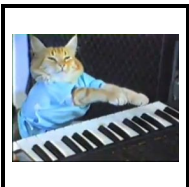
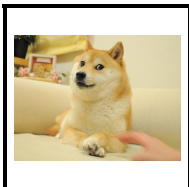

Image ICL Examples + Text Query				Output
			The logo is the letter P stylized to look like a pushpin.	<b>Text ICL Base:</b> <span style="background-color: #d9ead3;">Pinterest</span> <b>Image ICL xBase:</b> <span style="background-color: #d9ead3;">Mapquest</span> <b>Image ICL xPatch:</b> <span style="background-color: #d9ead3;">Pinterest.</span>
Apple	Snapchat	Instagram	?	
			The character is a pink starfish wearing green and purple pants.	<b>Text ICL Base:</b> <span style="background-color: #d9ead3;">SpongeBob</span> <b>Image ICL xBase:</b> <span style="background-color: #d9ead3;">Plankton</span> <b>Image ICL xPatch:</b> <span style="background-color: #d9ead3;">Patrick Star.</span>
Sandy Cheeks	Mrs. Puff	Mr. Krabs	?	
			An image of an unhappy cat with blue eyes and white and brown fur.	<b>Text ICL Base:</b> <span style="background-color: #d9ead3;">Garfield</span> <b>Image ICL xBase:</b> <span style="background-color: #d9ead3;">Grumpy Cat</span> <b>Image ICL xPatch:</b> <span style="background-color: #d9ead3;">Grumpy Cat</span>
Keyboard Cat	Doge	This Is Fine Dog	?	

Figure 10: **Transfer from image ICL to text queries.** We show qualitative examples where few-shot prompting with text ICL (Base) and image ICL (xBase) often produces incorrect predictions in the same output domain while cross-modal patching (xPatch) leads to the correct answer.

We observe that global vector patching is often able to override local prompting but also fails when the task to patch is more challenging than the one provided in the same context window. For example, tasks like object recognition, color identification, or OCR that are highly emphasized in VLM training can be considered less challenging than a long-tail task like car logo recognition.

### 3.3 IMAGE ICL TRANSFER

Now we assess the usefulness of task vectors derived from image ICL for text queries, as originally formulated in Sec. 2.2. In Figure 10 we depict a set of tasks that involve recognizing visual concepts in dense textual descriptions, including mapping the description to a technology company, cartoon character, or popular meme. We provide the text ICL descriptions in Sec. A.6 of the Appendix.

Similar to Sec. 3.1, the model struggles when cross-modal examples are applied via few-shot prompting (Image ICL xBase) but performs well when the same examples are patched as a task vector (Image ICL xPatch). Both baselines (Text ICL Base, Image ICL xBase) sometimes generate incorrect answers within the same output domain, suggesting that, rather than focusing on the input-output relationship, the model may be ignoring the input image or description. However, on the evaluation tasks in Table 3, it is difficult for image ICL to surpass the strong unimodal baselines. In Table 10 of the Appendix we include an ablation containing all possible combinations of specification-query modality for task vector patching, where text ICL consistently outperforms image ICL regardless of the query modality. We hypothesize that this phenomenon can be attributed to the nature of the tasks themselves. In the evaluation tasks, image ICL also has to complete an implicit recognition task mapping the image to the underlying textual concept. For example, if the model cannot match the flag to the correct country name, it will not be able to predict the correct currency. However, if recognition is instead required in text space, as is the case in Figure 10, image ICL may better encode the task. We think that the curation of a comprehensive evaluation set containing dense text descriptions and corresponding visual concepts is an exciting future direction.

## 4 RELATED WORK

**Mechanistic Interpretability.** The goal of mechanistic interpretability in deep learning is to make deep models more transparent and interpretable by understanding how and why model decisions are made (Gilpin et al., 2018; Gurnee & Tegmark; Liu et al., 2022; Geva et al., 2020; Nanda et al., 2023). To uncover the relationships within the model, *causal interventions* (Pearl, 2022) are often used. For example, Activation Patching (Zhang & Nanda, 2023) is a technique used to modify neural network activations to observe changes in outputs, often with causal insights to correct biased or erroneous behavior (Meng et al., 2022; Bau et al.). Here, we use Activation Patching to demonstrate that task representations transfer across modalities, regardless of being specified by examples or instructions.

**In Context Learning.** With the recent advent of LLMs (Brown et al., 2020), researchers have sought to explain in-context learning (Liu et al., 2023b), the phenomenon in which LLMs can adapt to new tasks with a few input examples in the forward pass. Olsson et al. (2022) hypothesized that ICL is driven by attention heads (“induction heads”), while Xie et al. (2021) interprets ICL as implicit Bayesian Inference process, and Garg et al. (2022) showed that ICL can emerge in the simple case of linear functions. More recently, Hendel et al. (2023) and Todd et al. (2024) hypothesized that ICL creates task (or function) vectors, latent activations that encode the task in LLMs, and Hojel et al. (2024) demonstrated a similar behavior in computer vision models. Huang et al. (2024) proposed to use task vectors in VLMs to compress long prompts that would otherwise not fit in a limited context length. We study how task information evolves within VLMs, specifically the similarity and transferability of the representation when the task is expressed in different modalities.

**Vision-and-Language Models.** Inspired by the success of LLMs, new vision-and-language models (VLMs) have been proposed (Liu et al., 2023a; Li et al., 2023; Tong et al., 2024; Team, 2024; Laurençon et al., 2024; Zhou et al., 2024). Recent VLMs can be roughly categorized to modality late-fusion (Liu et al., 2023a; 2024) and early-fusion (Bavishi et al., 2023; Lu et al., 2022; 2023; Team, 2024) approaches. Late-fusion approaches typically combine a pre-trained visual encoder and LLM by training adapters, potentially with a short end-to-end fine-tuning stage. In contrast, early-fusion approaches focus on end-to-end training without any pre-initialization of the representations. We observe cross-modal task representations for both model categories, suggesting that this property can emerge regardless of the initialization. Several works examine image ICL in VLMs, proposing new models designed for ICL (Alayrac et al., 2022; Laurençon et al., 2024; Doveh et al., 2024; Jiang et al., 2024) and analyzing the impact of in-context example selection on performance (Baldassini et al., 2024). Our work offers a new perspective on image ICL by comparing it with text ICL and demonstrating the similarity between the two processes. We even show VLMs that lack image ICL capabilities (Liu et al., 2023a; Lin et al., 2023; Doveh et al., 2024) can still benefit from task vectors.

## 5 LIMITATIONS

In this work, we demonstrate that VLMs learn cross-modal task representations but we lack a definitive explanation for *why*. Empirical studies offer several hypotheses, such as the existence of isomorphic structures between language and other perceptual representation spaces (Abdou et al., 2021; Patel & Pavlick, 2022; Pavlick, 2023), or representational convergence from modeling the same underlying reality (Huh et al., 2024). Additionally, we observe quantitative improvements for text-to-image transfer but not image-to-text transfer, possibly because VLM training is more text-centric. However, we believe that learning task representations from visual data has its advantages, and we provide qualitative examples where image-to-text transfer proves beneficial.

## 6 CONCLUSION

Vision-and-language models (VLMs) are generalist models capable of solving a wide range of computer vision tasks by framing them as question answering problems in text. Despite their success, we lack a clear understanding of how they work. Our primary observation is that VLMs map inputs into a shared task representation space, regardless of whether the task is defined by text examples, image examples, or explicit instructions. Based on this, we show it is possible to transfer task vectors from one modality (e.g., text) to another (e.g., images). We hope our work will inspire further exploration into the inductive biases of VLMs and the reasons behind their success.

## REFERENCES

- 540  
541  
542 Mostafa Abdou, Artur Kulmizev, Daniel Hershcovich, Stella Frank, Ellie Pavlick, and Anders  
543 Søgaard. Can language models encode perceptual structure without grounding? a case study  
544 in color, 2021. URL <https://arxiv.org/abs/2109.06129>.
- 545 Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel  
546 Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, Roman Ring, Eliza Rutherford,  
547 Serkan Cabi, Tengda Han, Zhitao Gong, Sina Samangooei, Marianne Monteiro, Jacob Menick,  
548 Sebastian Borgeaud, Andrew Brock, Aida Nematzadeh, Sahand Sharifzadeh, Mikolaj Binkowski,  
549 Ricardo Barreira, Oriol Vinyals, Andrew Zisserman, and Karen Simonyan. Flamingo: a vi-  
550 sual language model for few-shot learning. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave,  
551 and Kyunghyun Cho (eds.), *Advances in Neural Information Processing Systems*, 2022. URL  
552 <https://openreview.net/forum?id=EbMuimAbPbs>.
- 553 Anthropic. Claude 3.5 sonnet, 2024. URL <https://www.anthropic.com/news/claude-3-5-sonnet>.
- 554  
555 Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C. Lawrence  
556 Zitnick, and Devi Parikh. VQA: Visual Question Answering. In *International Conference on*  
557 *Computer Vision (ICCV)*, 2015.
- 559 Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge,  
560 Yu Han, Fei Huang, Binyuan Hui, Luo Ji, Mei Li, Junyang Lin, Runji Lin, Dayiheng Liu, Gao Liu,  
561 Chengqiang Lu, Keming Lu, Jianxin Ma, Rui Men, Xingzhang Ren, Xuancheng Ren, Chuanqi  
562 Tan, Sinan Tan, Jianhong Tu, Peng Wang, Shijie Wang, Wei Wang, Shengguang Wu, Benfeng  
563 Xu, Jin Xu, An Yang, Hao Yang, Jian Yang, Shusheng Yang, Yang Yao, Bowen Yu, Hongyi  
564 Yuan, Zheng Yuan, Jianwei Zhang, Xingxuan Zhang, Yichang Zhang, Zhenru Zhang, Chang  
565 Zhou, Jingren Zhou, Xiaohuan Zhou, and Tianhang Zhu. Qwen technical report, 2023a. URL  
566 <https://arxiv.org/abs/2309.16609>.
- 567 Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang  
568 Zhou, and Jingren Zhou. Qwen-vl: A versatile vision-language model for understanding, local-  
569 ization, text reading, and beyond, 2023b. URL <https://arxiv.org/abs/2308.12966>.
- 570 Folco Bertini Baldassini, Mustafa Shukor, Matthieu Cord, Laure Soulier, and Benjamin Piwowarski.  
571 What makes multimodal in-context learning work?, 2024.
- 572  
573 Anthony Bau, Yonatan Belinkov, Hassan Sajjad, Nadir Durrani, Fahim Dalvi, and James Glass.  
574 Identifying and controlling important neurons in neural machine translation. In *International*  
575 *Conference on Learning Representations*.
- 576 Rohan Bavishi, Erich Elsen, Curtis Hawthorne, Maxwell Nye, Augustus Odena, Arushi Somani,  
577 and Sagnak Tasirlar. Fuyu-8b: A multimodal architecture for ai agents, 2023. URL <https://www.adept.ai/blog/fuyu-8b>.
- 578  
579 Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal,  
580 Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are  
581 few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.
- 582  
583 Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng,  
584 Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. Vicuna: An  
585 open-source chatbot impressing gpt-4 with 90%\* chatgpt quality, March 2023. URL <https://lmsys.org/blog/2023-03-30-vicuna/>.
- 586  
587 Sivan Doherty, Shaked Perek, M. Jehanzeb Mirza, Wei Lin, Amit Alfassy, Assaf Arbel, Shimon  
588 Ullman, and Leonid Karlinsky. Towards multimodal in-context learning for vision & language  
589 models, 2024. URL <https://arxiv.org/abs/2403.12736>.
- 590  
591 Shivam Garg, Dimitris Tsipras, Percy S Liang, and Gregory Valiant. What can transformers learn  
592 in-context? a case study of simple function classes. *Advances in Neural Information Processing*  
593 *Systems*, 35:30583–30598, 2022.

- 594 Mor Geva, Roei Schuster, Jonathan Berant, and Omer Levy. Transformer feed-forward layers are  
595 key-value memories. *arXiv preprint arXiv:2012.14913*, 2020.  
596
- 597 Leilani H Gilpin, David Bau, Ben Z Yuan, Ayesha Bajwa, Michael Specter, and Lalana Kagal.  
598 Explaining explanations: An overview of interpretability of machine learning. In *2018 IEEE*  
599 *5th International Conference on data science and advanced analytics (DSAA)*, pp. 80–89. IEEE,  
600 2018.
- 601 Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. Making the V  
602 in VQA matter: Elevating the role of image understanding in Visual Question Answering. In  
603 *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.  
604
- 605 Wes Gurnee and Max Tegmark. Language models represent space and time. In *The Twelfth Inter-*  
606 *national Conference on Learning Representations*.
- 607 Roei Hendel, Mor Geva, and Amir Globerson. In-context learning creates task vectors. *Findings of*  
608 *Empirical Methods in Natural Language Processing*, 2023.  
609
- 610 Alberto Hojel, Yutong Bai, Trevor Darrell, Amir Globerson, and Amir Bar. Finding visual task  
611 vectors. *European Conference on Computer Vision*, 2024.
- 612 Brandon Huang, Chancharik Mitra, Assaf Arbelle, Leonid Karlinsky, Trevor Darrell, and Roei  
613 Herzig. Multimodal task vectors enable many-shot multimodal in-context learning. *arXiv preprint*  
614 *arXiv:2406.15334*, 2024.  
615
- 616 Minyoung Huh, Brian Cheung, Tongzhou Wang, and Phillip Isola. The platonic representation  
617 hypothesis. In *ICML*, 2024.  
618
- 619 Gabriel Ilharco, Mitchell Wortsman, Ross Wightman, Cade Gordon, Nicholas Carlini, Rohan Taori,  
620 Achal Dave, Vaishaal Shankar, Hongseok Namkoong, John Miller, Hannaneh Hajishirzi, Ali  
621 Farhadi, and Ludwig Schmidt. Openclip, July 2021. URL [https://doi.org/10.5281/](https://doi.org/10.5281/zenodo.5143773)  
622 [zenodo.5143773](https://doi.org/10.5281/zenodo.5143773). If you use this software, please cite it as below.
- 623 iNaturalist. inaturalist 2017 species classification and detection dataset. [https://github.com/](https://github.com/visipedia/inat_comp/tree/master/2017)  
624 [visipedia/inat\\_comp/tree/master/2017](https://github.com/visipedia/inat_comp/tree/master/2017), 2017.  
625
- 626 Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chap-  
627 lot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier,  
628 L elio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril,  
629 Thomas Wang, Timoth ee Lacroix, and William El Sayed. Mistral 7b, 2023. URL [https://](https://arxiv.org/abs/2310.06825)  
630 [arxiv.org/abs/2310.06825](https://arxiv.org/abs/2310.06825).
- 631 Dongfu Jiang, Xuan He, Huaye Zeng, Cong Wei, Max Ku, Qian Liu, and Wenhui Chen. Mantis:  
632 Interleaved multi-image instruction tuning, 2024. URL [https://arxiv.org/abs/2405.](https://arxiv.org/abs/2405.01483)  
633 [01483](https://arxiv.org/abs/2405.01483).
- 634 Hugo Lauren¸con, L eo Tronchon, Matthieu Cord, and Victor Sanh. What matters when building  
635 vision-language models?, 2024. URL <https://arxiv.org/abs/2405.02246>.  
636
- 637 Bo Li, Hao Zhang, Kaichen Zhang, Dong Guo, Yuanhan Zhang, Renrui Zhang, Feng  
638 Li, Ziwei Liu, and Chunyuan Li. Llava-next: What else influences visual instruc-  
639 tion tuning beyond data?, May 2024. URL [https://llava-vl.github.io/blog/](https://llava-vl.github.io/blog/2024-05-25-llava-next-ablations/)  
640 [2024-05-25-llava-next-ablations/](https://llava-vl.github.io/blog/2024-05-25-llava-next-ablations/).
- 641 Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image  
642 pre-training with frozen image encoders and large language models. In *International conference*  
643 *on machine learning*, pp. 19730–19742. PMLR, 2023.  
644
- 645 Weixin Liang, Lili Yu, Liang Luo, Srinivasan Iyer, Ning Dong, Chunting Zhou, Gargi Ghosh, Mike  
646 Lewis, Wen tau Yih, Luke Zettlemoyer, and Xi Victoria Lin. Mixture-of-transformers: A sparse  
647 and scalable architecture for multi-modal foundation models, 2024. URL [https://arxiv.](https://arxiv.org/abs/2411.04996)  
[org/abs/2411.04996](https://arxiv.org/abs/2411.04996).

- 648 Ji Lin, Hongxu Yin, Wei Ping, Yao Lu, Pavlo Molchanov, Andrew Tao, Huizi Mao, Jan Kautz,  
649 Mohammad Shoeybi, and Song Han. Vila: On pre-training for visual language models, 2023.  
650
- 651 Ji Lin, Hongxu Yin, Wei Ping, Yao Lu, Pavlo Molchanov, Andrew Tao, Huizi Mao, Jan Kautz,  
652 Mohammad Shoeybi, and Song Han. Vila: On pre-training for visual language models. In *Pro-*  
653 *ceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*,  
654 2024.
- 655 Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning, 2023a.  
656
- 657 Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction  
658 tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recogni-*  
659 *tion*, pp. 26296–26306, 2024.
- 660 Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. Pre-  
661 train, prompt, and predict: A systematic survey of prompting methods in natural language pro-  
662 cessing. *ACM Computing Surveys*, 55(9):1–35, 2023b.
- 663 Ziming Liu, Ouail Kitouni, Niklas S Nolte, Eric Michaud, Max Tegmark, and Mike Williams. To-  
664 wards understanding grokking: An effective theory of representation learning. *Advances in Neu-*  
665 *ral Information Processing Systems*, 35:34651–34663, 2022.  
666
- 667 Jiasen Lu, Christopher Clark, Rowan Zellers, Roozbeh Mottaghi, and Aniruddha Kembhavi.  
668 Unified-io: A unified model for vision, language, and multi-modal tasks, 2022. URL <https://arxiv.org/abs/2206.08916>.  
669
- 670 Jiasen Lu, Christopher Clark, Sangho Lee, Zichen Zhang, Savva Khosla, Ryan Marten, Derek  
671 Hoiem, and Aniruddha Kembhavi. Unified-io 2: Scaling autoregressive multimodal models with  
672 vision, language, audio, and action, 2023. URL <https://arxiv.org/abs/2312.17172>.  
673
- 674 Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. Locating and editing factual  
675 associations in gpt. *Advances in Neural Information Processing Systems*, 35:17359–17372, 2022.
- 676 Neel Nanda, Lawrence Chan, Tom Lieberum, Jess Smith, and Jacob Steinhardt. Progress measures  
677 for grokking via mechanistic interpretability. *arXiv preprint arXiv:2301.05217*, 2023.  
678
- 679 nostalgebraist. interpreting gpt: the logit lens. LessWrong, 2020.  
680 URL [https://www.lesswrong.com/posts/AcKRB8wDpdaN6v6ru/](https://www.lesswrong.com/posts/AcKRB8wDpdaN6v6ru/interpreting-gpt-the-logit-lens)  
681 [interpreting-gpt-the-logit-lens](https://www.lesswrong.com/posts/AcKRB8wDpdaN6v6ru/interpreting-gpt-the-logit-lens).
- 682 Catherine Olsson, Nelson Elhage, Neel Nanda, Nicholas Joseph, Nova DasSarma, Tom Henighan,  
683 Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, et al. In-context learning and induction  
684 heads. *arXiv preprint arXiv:2209.11895*, 2022.  
685
- 686 Roma Patel and Ellie Pavlick. Mapping language models to grounded conceptual spaces. In *Interna-*  
687 *tional Conference on Learning Representations*, 2022. URL [https://openreview.net/](https://openreview.net/forum?id=gJcEM8sxHK)  
688 [forum?id=gJcEM8sxHK](https://openreview.net/forum?id=gJcEM8sxHK).
- 689 Ellie Pavlick. Symbols and grounding in large language models. *Philosophical Transactions of*  
690 *the Royal Society A*, 381(20220041), 2023. doi: 10.1098/rsta.2022.0041. URL [http://doi.](http://doi.org/10.1098/rsta.2022.0041)  
691 [org/10.1098/rsta.2022.0041](http://doi.org/10.1098/rsta.2022.0041).
- 692 Judea Pearl. Direct and indirect effects. In *Probabilistic and causal inference: the works of Judea*  
693 *Pearl*, pp. 373–392. 2022.  
694
- 695 Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language  
696 models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.
- 697 Chameleon Team. Chameleon: Mixed-modal early-fusion foundation models. *arXiv preprint*  
698 *arXiv:2405.09818*, 2024.  
699
- 700 Eric Todd, Millicent L Li, Arnab Sen Sharma, Aaron Mueller, Byron C Wallace, and David Bau.  
701 Function vectors in large language models. *International Conference on Learning Representa-*  
*tions*, 2024.

702 Shengbang Tong, Ellis Brown, Penghao Wu, Sanghyun Woo, Manoj Middepogu, Sai Charitha  
703 Akula, Jihan Yang, Shusheng Yang, Adithya Iyer, Xichen Pan, et al. Cambrian-1: A fully open,  
704 vision-centric exploration of multimodal llms. *arXiv preprint arXiv:2406.16860*, 2024.  
705

706 Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of Ma-*  
707 *chine Learning Research*, 9(86):2579–2605, 2008. URL [http://jmlr.org/papers/v9/](http://jmlr.org/papers/v9/vandermaaten08a.html)  
708 [vandermaaten08a.html](http://jmlr.org/papers/v9/vandermaaten08a.html).

709 Sang Michael Xie, Aditi Raghunathan, Percy Liang, and Tengyu Ma. An explanation of in-context  
710 learning as implicit bayesian inference. *arXiv preprint arXiv:2111.02080*, 2021.  
711

712 Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language  
713 image pre-training, 2023. URL <https://arxiv.org/abs/2303.15343>.

714 Fred Zhang and Neel Nanda. Towards best practices of activation patching in language models:  
715 Metrics and methods. *arXiv preprint arXiv:2309.16042*, 2023.  
716

717 Chunting Zhou, Lili Yu, Arun Babu, Kushal Tirumala, Michihiro Yasunaga, Leonid Shamis, Jacob  
718 Kahn, Xuezhe Ma, Luke Zettlemoyer, and Omer Levy. Transfusion: Predict the next token and  
719 diffuse images with one multi-modal model. *arXiv preprint arXiv:2408.11039*, 2024.  
720  
721  
722  
723  
724  
725  
726  
727  
728  
729  
730  
731  
732  
733  
734  
735  
736  
737  
738  
739  
740  
741  
742  
743  
744  
745  
746  
747  
748  
749  
750  
751  
752  
753  
754  
755