SHARCS: Shared Concept Space for Explainable Multimodal Learning

Gabriele Dominici
Università della Svizzera Italiana
Lugano, Switzerland
gabriele-dominici@usi.ch

Pietro Barbiero
Università della Svizzera Italiana
Lugano, Switzerland
University of Cambridge
Cambridge, UK
pietro-barbiero@usi.ch

Lucie Charlotte Magister
University of Cambridge
Cambridge, UK
lcm67@cam.ac.uk

Pietro Liò
University of Cambridge
Cambridge, UK
pl219@cam.ac.uk

Nikola Simidjievski
University of Cambridge
Cambridge, UK
ns779@cam.ac.uk

Abstract

Multimodal learning is an essential paradigm for addressing complex real-world problems, where individual data modalities are typically insufficient for accurately solving a given modelling task. While various deep learning approaches have successfully addressed these challenges, their reasoning process is often opaque; limiting the capabilities for a principled explainable cross-modal analysis and any domain-expert intervention. In this paper, we introduce SHARCS (SHARed Concept Space) – a novel concept-based approach for explainable multimodal learning. SHARCS learns and maps interpretable concepts from different heterogeneous modalities into a single unified concept-manifold, which leads to an intuitive projection of semantically similar cross-modal concepts. We demonstrate that such an approach can lead to inherently explainable task predictions while also improving downstream predictive performance. Moreover, we show that SHARCS can operate and significantly outperform other approaches in practically significant scenarios, such as retrieval of missing modalities and cross-modal explanations. Our approach is model agnostic and easily applicable to different types (and number) of modalities, thus advancing the development of effective, interpretable, and trustworthy multimodal approaches.

1 Introduction

Deep learning (DL) approaches for multimodal learning attain high performance by blending information from different data sources [22, 14]. However, the opaque reasoning of DL models [24] hinders the human ability to better understand the relationships in the data across modalities, which is imperative for safety-critical domains such as healthcare and biology, where this may often lead to novel insights and discoveries. To address this issue, many self-explainable methods were released [13, 28, 1, 2], offering an effective solution to bridge this knowledge gap. These methods can extract intuitive and human-readable explanations, and some even facilitate interaction with human experts, enabling a deeper understanding of the problem. However, they are often limited to single data modalities. A recent line of research focuses explicitly on developing or adapting existing methods for multimodal settings [23]. While relevant, they are typically tailored for specific multimodal scenar-

Preprint. Under review.

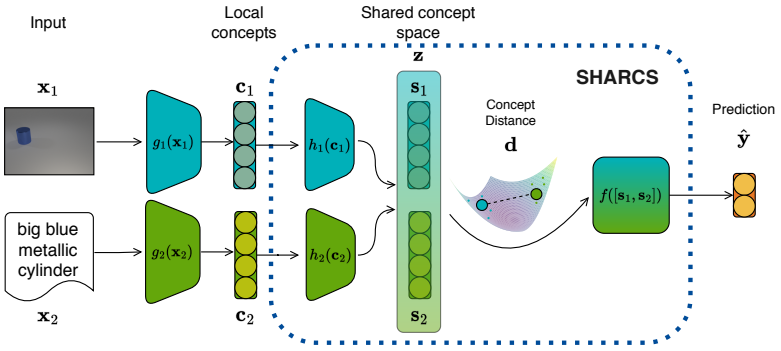


Figure 1: **SHARCS (SHARed Concept Space)**: for each modality i, the concept encoder module g_i produces a local concept embedding $\mathbf{c}_i$. SHARCS then maps local concept embeddings into a shared concept representation $\mathbf{s}_i$. To generate a semantically meaningful shared space, SHARCS minimises the distance between shared concepts of similar objects from different modalities. Finally, the label predictor f takes as input the concatenation of all shared concepts $\mathbf{s_i}$ to solve the task at hand.

ios [26], provide only local explanations [19, 15] or generate explanations for just one of the modalities [11] using an extra modality, thus failing to provide a general solution to multimodal problems.

In this paper, we introduce SHARCS (SHARed Concept Space), a novel interpretable concept-based approach (described in Section 2) designed to address general multimodal tasks. Our experiments (Section 3) demonstrate on four common data modalities (tabular, text, image, and graph data) that SHARCS (i) outperforms unimodal models and matches the task performance of existing baselines on challenging multimodal settings, (ii) attains high task accuracy even when a modality is missing, (iii) generates intuitive concept-based explanations for task predictions, and (iv) generates simple concept-based explanations for a data modality using the concepts emerging from other modalities, allowing human experts to uncover hidden cross-modal connections.

2 SHARCS: SHARed Concepts Space

SHARCS combines information from diverse data sources during training, emphasizing the integration of high-level, interpretable concept representations (as defined by Ghorbani et al. [7]), as opposed to traditional uninterpretable embeddings [24]. This approach facilitates intuitive concept-based explanations and enables experts to explore the interrelationships between data modalities. In SHARCS, for example, a red ball is represented as a multimodal concept with a consistent representation in the shared space across input modalities (e.g., image, text, etc.).

Local concepts Figure 1 depicts SHARCS applied to two data modalities. The model utilizes concept encoders $g_1, \ldots, g_n$, each for a modality $i = 1, \ldots, n$, mapping inputs to local concepts. Modality-specific architectures $\phi_1, \ldots, \phi_n$ map inputs to latent concept representations. To convert latent concepts into a local concept space, we use batch scaling $\circledast : \mathbb{R}^{b \times k} \to \mathbb{R}^k$ (with batch size b) and a sigmoid activation function $\sigma : \mathbb{R} \to [0, 1]$, resulting in $g_i = \sigma \circ \circledast \circ \phi_i$. Batch rescaling before sigmoid activation triggers a concept when its representation significantly differs from others in the batch:

$$\mathbf{c}_{i,m} = \sigma\Big(\phi_i(\mathbf{x}_{i,m}) \underset{j \in B_{i,m}}{\circledast} \phi_i(\mathbf{x}_{i,j})\Big)^{-1} \tag{1}$$

Here, $B_{i,m}$ is the m-th batch's sample indexes, $\circledast$ represents permutation-invariant batch rescaling (e.g., batch normalization), and $\mathbf{c}_i$ decipts local concepts for the i-th modality. They can be used to understand how local concepts combine into the shared concept space, offering another level of interpretability of the model.

Shared concepts SHARCS then maps the local concepts $\mathbf{c}_i$ into a shared concept space. To this end, SHARCS applies a modality-specific set of concept encoders $h_1, \ldots, h_n$ mapping local concepts $\mathbf{c}_i \in C \subseteq [0, 1]^k$ into a set of shared concept embeddings $\mathbf{s}_i \in S \subseteq [0, 1]^t$ of size t i.e., $h_i : C_i \to S$. Shared concept encoders resemble the structure of local encoders applying batch rescaling and a

sigmoid activation on top of learnable parametric functions $\psi_1, \ldots, \psi_n$:

$$\mathbf{s}_{i,m} = \sigma\Big(\psi_i(\mathbf{c}_{i,m}) \underset{i=1,\ldots,n \,\wedge\, j \in B_{i,m}}{\circledast} \psi_i(\mathbf{c}_{i,j})\Big)^{-1} \tag{2}$$

Thanks to this operation, our model blends information from different data modalities into the same space, enabling the generation of unified concept manifolds.

Task prediction. Finally, the model concatenates the shared concepts $\mathbf{s}_i$ from each modality and uses them to solve the task at hand. To solve the task, a label predictor function $f : S^n \to Y$ maps the shared concepts to a downstream task space $Y \subseteq \mathbb{R}^l$: $\hat{\mathbf{y}}_m = f\big(\mathbf{s}_{1,m} | \ldots | \mathbf{s}_{n,m}\big)$ where the symbol $|$ represents the concatenation operation. We provide more details about SHARCS in Appendix A.

Learning process SHARCS integrates information from diverse data modalities into a unified vector space. However, concept encoders can learn different concepts for various tasks, potentially causing overlap in the shared vector space. To address this, we introduce an additional term in the loss function $\mathcal{L}$, promoting semantic coherence by connecting concepts from different modalities:

$$\mathcal{L}(\mathbf{y}, \hat{\mathbf{y}}, \mathbf{s}) = \mathcal{T}(\mathbf{y}, \hat{\mathbf{y}}) + \frac{\lambda}{|M|} \sum_{(i,q) \in M \subseteq \binom{\{1,\ldots,n\}}{2}} \big|\big|\mathbf{s}_i - \mathbf{s}_q\big|\big|_2 \tag{3}$$

where $\lambda \in \mathbb{R}$ is a hyperparameter that controls the strength of our semantic regularization, $\mathcal{T}$ is a task-specific loss function (such as cross-entropy), M is a subset of all possible pairs of modalities $\binom{\{1,\ldots,n\}}{2}$, and $||\mathbf{s}_i - \mathbf{s}_q||_2$ represents the Euclidean norm between the shared representation of the same sample in the two different modalities i and q. In our solution, we randomly draw samples to compute the semantic regularisation loss at every iteration. Notably, a SHARCS-based model is flexible and can be configured differently, such as adding local tasks with modality-specific loss functions or employing various learning mechanisms like End-to-end, Sequential, and Local pre-training (cf. Appendix A.2).

Unimodal and Multimodal Explanations: SHARCS offers concept-based explanations, distinguishing it from existing multimodal models. Like unimodal unsupervised concept-based models [7], SHARCS assigns semantic meaning to concepts through visualization. It does this by highlighting examples with the highest or lowest concept values (concept active or inactive) in the shared space, eliminating the need for external algorithms. Prototypes of a concept can be retrieved in modality i by selecting samples with the highest (lowest) shared concept representation during training. Moreover, SHARCS can offer semantic context for an input sample by identifying the samples in the shared space that share the same concepts. This visualization helps identify relevant sample clusters with shared characteristics, providing insights into why these examples are classified similarly.

Cross-Modal Explanations & handling missing modalities: SHARCS goes beyond unimodal interpretability by enabling cross-modal explanations. Using an input sample from one modality, it retrieves similar examples from other modalities based on shared concept space proximity. Additionally, SHARCS can visualize how a concept in one modality is interpreted in another by displaying samples with the highest (lowest) values of that concept across all shared spaces. This further translates to another benefit: SHARCS can effectively handles missing modalities by approximating the representation from a reference modality. It identifies the closest shared concept in the reference modality and approximates the missing representation. This allows SHARCS to process inputs with missing modalities efficiently. Additional details on each of these properties can be found in the Appendix A.1

3 Experiments

Our central hypothesis is that SHARCS allows for an efficient, accurate and interpretable multimodal learning. To address these aspects, we design our experiments along two main points: i) Multimodal generalisation performance - Through a series of experiments, we first evaluate SHARCS' capabilities for multimodal learning in different practically-relevant scenarios. Then, we compare SHARCS performance to unimodal and multimodal baselines, some of which are not interpretable; ii) Interpretability - We qualitatively showcase SHARCS capabilities for learning semantically plausible, explainable and consistent (multimodal) concepts.

We evaluate our hypotheses on four multimodal tasks, each leveraging a pair of multimodal datasets such as tabular, image, graph, and text data. The four multimodal, or global, tasks are designed such

that the models need to leverage both modalities in order to provide correct predictions. Models that will learn only from one of the modalities will be able to solve a partial (local) single-modality task but will typically exhibit random performance on the global multimodal task. Furthermore, we test the interpretability of SHARCS and its ability to cope with real-world scenarios when a modality is missing. Further details of the experiments, dataset and model details, baselines and metrics used are in the Appendix B.

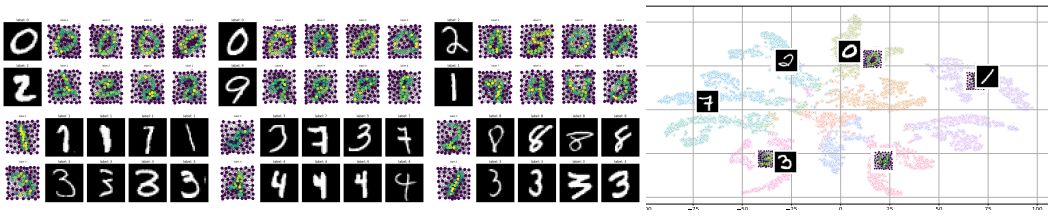
4 Results and discussion

SHARCS' generalisation is on par with non-interpretable multimodal models. As a proof-of-principle of our method, we first benchmarked it against unimodal approaches (see Figure 7 in the Appendix D). SHARCS achieves good performance across all four multimodal tasks, consistently outperforming (up to 81%) the unimodal baselines. Furthermore, the results presented in Table 1 show that SHARCS achieves slightly better or comparable performance than the other multimodal baselines. In particular, our approach can maintain good performance, despite the bottleneck introduced for computing concepts and the constraint of the shared space. More importantly, both concept-based approaches are the only two that can accurately model the CLEVR task, which further justifies the utility of the concept embeddings. Moreover, in terms of performance in scenarios with missing modalities, SHARCS consistently outperforms other baselines in Table 1. Its success lies in constructing a more robust and less noisy concept space, enhancing sample representation and enabling precise retrieval of missing modalities.

SHARCS unveils meaningful concepts and. SHARCS, much like our Concept Multimodal baselines, excels at extracting task-related concepts, evidenced by its completeness score [27] aligning closely with Accuracy in Table 1. Notably, SHARCS achieves higher completeness scores on three of the four datasets compared to solutions lacking a shared space, with improvements of up to 10% in MNIST+Superpixels. SHARCS uses the shared space to de-noise concepts, collapsing less significant ones into semantically richer representations. Additionally, SHARCS can offers insights into the prediction process by replacing the predictor function f with a decision tree (see Figures 12 and 13 in Appendix D). This enables users to understand how various concepts contribute to decisions, en-

Table 1: The performance of SHARCS (Accuracy (%)) in scenarios with missing modalities, compared to Relative representation and Concept Multimodal variants. The global task accuracy is presented as a reference. SHARCS performs better than the baselines, particularly on harder tasks requiring both modalities. In some scenarios, SHARCS is able to retrieve modalities, leading to better downstream performance than the original data.

		Global		Missing Modality	
Dataset	**Model**	**Accuracy**	**Compl.**	**1st Modality**	**2nd Modality**
XOR-AND-XOR	Relative	**99.5** $\pm$ 0.3	-	80.1 $\pm$ 6.4	82.8 $\pm$ 2.2
	Concept	99.0 $\pm$ 0.8	96.2 $\pm$ 1.2	68.0 $\pm$ 2.0	57.0 $\pm$ 6.1
	SHARCS	98.7 $\pm$ 0.5	**98.0** $\pm$ 1.2	**98.6** $\pm$ 0.9	**91.9** $\pm$ 1.2
MNIST+SuperP.	Relative	80.4 $\pm$ 0.2	-	52.6 $\pm$ 4.9	30.1 $\pm$ 2.4
	Concept	88.2 $\pm$ 0.1	78.9 $\pm$ 1.4	13.7 $\pm$ 3.9	10.8 $\pm$ 2.6
	SHARCS	**89.6** $\pm$ 0.1	**88.7** $\pm$ 0.2	**98.0** $\pm$ 0.0	**82.5** $\pm$ 0.4
HalfMNIST	Relative	**95.6** $\pm$ 0.1	-	92.9 $\pm$ 1.4	**60.1** $\pm$ 3.4
	Concept	93.9 $\pm$ 0.0	91.3 $\pm$ 0.1	89.4 $\pm$ 1.3	13.4 $\pm$ 2.1
	SHARCS	94.0 $\pm$ 0.1	**92.6** $\pm$ 0.3	**96.5** $\pm$ 0.0	55.1 $\pm$ 3.0
CLEVR	Relative	48.7 $\pm$ 0.5	-	49.9 $\pm$ 0.0	49.0 $\pm$ 0.1
	Concept	90.1 $\pm$ 1.0	**82.3** $\pm$ 1.2	51.4 $\pm$ 2.8	48.6 $\pm$ 2.7
	SHARCS	**90.2** $\pm$ 0.2	81.5 $\pm$ 1.1	**93.1** $\pm$ 0.6	**93.4** $\pm$ 0.4



(a) (b) (c) (d)

Figure 2: (a-c) Retrieval examples obtained by (a) SHARCS, (b) Relative representation, and (c) Concept Multimodal; on the MNIST+Superpixels dataset. The top two rows are samples of retrieved graphs using images, while the bottom two are retrieved images using graph samples. (d) tSNE plot of the SHARCS concept space

hancing task comprehension, revealing how concepts combine, and justifying sample classifications. Moreover, **SHARCS enhance cross-modal understanding.** This cross-modal explanation can be extended to individual concepts, demonstrating how specific concepts are represented in the other modality. Additionally, Figure 2d shows SHARCS' shared space for the MNIST+Superpixels dataset, where similar examples from different modalities are closely mapped. This property is extremely valuable, particularly when modalities lack expressiveness or share nuanced commonalities, shedding light on the critical relationship between modalities and samples, which can be beneficial across domains like medicine, biology, and healthcare. We envision this work as a foundation for the development and evaluation of interpretable multimodal approaches.

5 Acknowledgments

This study was funded by TRUST-ME (project 205121L_214991) and SmartCHANGE (GA No. 101080965) projects.

References

[1] David Alvarez-Melis and Tommi S. Jaakkola. Towards robust interpretability with self-explaining neural networks, 2018.

[2] Chaofan Chen, Oscar Li, Daniel Tao, Alina Barnett, Cynthia Rudin, and Jonathan K Su. This looks like that: deep learning for interpretable image recognition. *Advances in neural information processing systems*, 32, 2019.

[3] Li Deng. The mnist database of handwritten digit images for machine learning research. *IEEE Signal Processing Magazine*, 29(6):141–142, 2012.

[4] Inderjit S. Dhillon, Yuqiang Guan, and Brian Kulis. Weighted graph cuts without eigenvectors a multilevel approach. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29 (11):1944–1957, 2007. doi: 10.1109/TPAMI.2007.1115.

[5] Matthias Fey and Jan E. Lenssen. Fast graph representation learning with PyTorch Geometric. In *ICLR Workshop on Representation Learning on Graphs and Manifolds*, 2019.

[6] Matthias Fey, Jan Eric Lenssen, Frank Weichert, and Heinrich Müller. Splinecnn: Fast geometric deep learning with continuous b-spline kernels, 2018.

[7] Amirata Ghorbani, Abubakar Abid, and James Zou. Interpretation of neural networks is fragile. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 3681–3688, 2019.

[8] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015.

[9] J. D. Hunter. Matplotlib: A 2d graphics environment. *Computing in Science & Engineering*, 9 (3):90–95, 2007. doi: 10.1109/MCSE.2007.55.

[10] Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax, 2017.

[11] Jinkyu Kim, Anna Rohrbach, Trevor Darrell, John Canny, and Zeynep Akata. Textual explanations for self-driving vehicles, 2018.

[12] Thomas N. Kipf and Max Welling. Semi-Supervised Classification with Graph Convolutional Networks. In *5th International Conference on Learning Representations, ICLR 2017 - Conference Track Proceedings*, pages 11313–11320. International Conference on Learning Representations, ICLR, sep 2016. URL `http://arxiv.org/abs/1609.02907`.

[13] Pang Wei Koh, Thao Nguyen, Yew Siang Tang, Stephen Mussmann, Emma Pierson, Been Kim, and Percy Liang. Concept bottleneck models. In *International Conference on Machine Learning*, pages 5338–5348. PMLR, 2020.

[14] Jie Lei, Linjie Li, Luowei Zhou, Zhe Gan, Tamara L. Berg, Mohit Bansal, and Jingjing Liu. Less is more: Clipbert for video-and-language learning via sparse sampling, 2021.

[15] Qing Li, Qingyi Tao, Shafiq Joty, Jianfei Cai, and Jiebo Luo. Vqa-e: Explaining, elaborating, and enhancing your answers for visual questions, 2018.

[16] Lucie Charlotte Magister, Pietro Barbiero, Dmitry Kazhdna, Federico Siciliano, Gabriele Ciravegna, Fabrizio Silvestri, Pietro Liò, and Mateja Jamnik. Encoding concepts in graph neural networks. *Advances in neural information processing systems*, 2022. [Under review].

[17] Federico Monti, Davide Boscaini, Jonathan Masci, Emanuele Rodolà, Jan Svoboda, and Michael M. Bronstein. Geometric deep learning on graphs and manifolds using mixture model cnns, 2016.

[18] Luca Moschella, Valentino Maiorca, Marco Fumero, Antonio Norelli, Francesco Locatello, and Emanuele Rodolà. Relative representations enable zero-shot latent space communication, 2023.

[19] Dong Huk Park, Lisa Anne Hendricks, Zeynep Akata, Anna Rohrbach, Bernt Schiele, Trevor Darrell, and Marcus Rohrbach. Multimodal explanations: Justifying decisions and pointing to the evidence, 2018.

[20] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019. URL `http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf`.

[21] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[22] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision, 2021.

[23] Nikolaos Rodis, Christos Sardianos, Georgios Th. Papadopoulos, Panagiotis Radoglou-Grammatikis, Panagiotis Sarigiannidis, and Iraklis Varlamis. Multimodal explainable artificial intelligence: A comprehensive review of methodological advances and future research directions, 2023.

[24] Cynthia Rudin. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence*, 1(5):206–215, 2019.

[25] Raeid Saqur and Karthik Narasimhan. Multimodal graph networks for compositional generalization in visual question answering. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 3070–3081. Curran Associates, Inc., 2020. URL `https://proceedings.neurips.cc/paper_files/paper/2020/file/1fd6c4e41e2c6a6b092eb13ee72bce95-Paper.pdf`.

[26] Cheng Wang, Haojin Yang, Xiaoyin Che, and Christoph Meinel. Concept-based multimodal learning for topic generation. In Xiangjian He, Suhuai Luo, Dacheng Tao, Changsheng Xu, Jie Yang, and Muhammad Abul Hasan, editors, *MultiMedia Modeling - 21st International Conference, MMM 2015, Sydney, NSW, Australia, January 5-7, 2015, Proceedings, Part I*, volume 8935 of *Lecture Notes in Computer Science*, pages 385–395. Springer, 2015. doi: 10.1007/978-3-319-14445-0_33. URL `https://doi.org/10.1007/978-3-319-14445-0_33`.

[27] Chih-Kuan Yeh, Been Kim, Sercan Arik, Chun-Liang Li, Tomas Pfister, and Pradeep Ravikumar. On completeness-aware concept-based explanations in deep neural networks. *Advances in Neural Information Processing Systems*, 33:20554–20565, 2020.

[28] Mateo Espinosa Zarlenga, Pietro Barbiero, Gabriele Ciravegna, Giuseppe Marra, Francesco Giannini, Michelangelo Diligenti, Zohreh Shams, Frederic Precioso, Stefano Melacci, Adrian Weller, Pietro Liò, and Mateja Jamnik. Concept embedding models. *Advances in neural information processing systems*, 2021. [Under review].

A Additional details of SHARCS

A.1 Multimodal concept-based explanations

Unimodal and multimodal explanations. The key advantage of SHARCS with respect to existing multimodal models is that it provides intuitive concept-based explanations. Similarly to unimodal unsupervised concept-based models [7], we can use SHARCS to assign semantic meaning to concept labels by visualizing the "prototypes" of a concept, represented by the examples with the highest values (concept active) or with the lowest values (concept inactive) for that concept in the shared space. Thanks to the interpretable architecture, SHARCS does not require an external algorithm to find these samples as opposed to post-hoc methods such as Ghorbani et al. [7]. More formally, we can retrieve a prototype $\gamma_v \in \{0,1\}^t$ of a concept v in a modality i by taking the sample with the SHARCS shared concept representation with the highest (lowest) value across all the s_i seen during training.

Moreover, SHARCS can also provide a semantic contextualisation for an input sample m by visualising the input samples whose embeddings are closer in the shared space embeddings to the given input. More formally, given a reference modality i we can identify the set of closest samples to the input m in a radius $\rho \in \mathbb{R}$ as follows:

$$E = \{j \in B_{train} \mid ||\mathbf{s}_{i,m} - \mathbf{s}_{i,j}||_2 < \rho\} \tag{4}$$

where $B_{train} \subseteq \mathbb{N}$ is the set of indexes of training samples, $E \subseteq B_{train}$ is the set of the closest samples in the shared space. This form of visualisation is often used to find relevant clusters of samples sharing some key characteristics. By showcasing examples from the input concept's family, SHARCS allows users to comprehend why these examples are classified similarly.

Both can be applied to global concepts, as they are the concatenation of the shared concepts. The only difference it to consider $\mathbf{z}_j = (\mathbf{s}_{1,j} | \dots | \mathbf{s}_{n,j}) \in [0,1]^{n \times t}$, instead of s_i.

Cross-modal explanations. SHARCS offers unique forms of explanations which go significantly beyond simple unimodal interpretability. Indeed, SHARCS enables cross-modal explanations, allowing one modality to be explained using another. Specifically, we can use an input sample in a specific modality to retrieve the most similar examples from other modalities. To this end, we can select training samples from the other modalities, which are closer in the shared concept space to the sample m being explained:

$$E = \{j \in B_{train},\ q \in \{1, \dots, n\} \mid ||\mathbf{s}_{i,m} - \mathbf{s}_{q,j}||_2 < \rho\} \tag{5}$$

As before, it is also possible to visualise how a concept v of a modality i is interpreted in the other modality q, visualising the samples with the highest (or lowest) value of the concept v in modality q across all $s_q v$. In such a manner, it is possible to translate a key feature from one modality to another.

These functionalities are particularly valuable when a modality's features are less human-interpretable than others. Visualizing the relationships between modalities enables cross-modal interpretability by emphasizing the semantic interconnections between concepts of different modalities.

Inference with missing modalities. Another unique feature of SHARCS is that the shared concept space enables it to process inputs with missing modalities effectively. Indeed, the original representation of an input m of a missing modality i can be effectively approximated using the shared concepts of another reference modality q. To this end, we just need to find the shared concept $\mathbf{s}_{i,j}$ observed during training from the missing modality, which is closest to a shared concept of the reference modality $m' = \arg\min_{j \in B_{train}} ||\mathbf{s}_{q,m} - \mathbf{s}_{i,j}||_2$

This way we can approximate the missing shared concept representation $\mathbf{s}_{im}$ from the missing modality as follows:

$$\mathbf{s}_{i,m'} = \sigma\Big(\psi_i(\mathbf{c}_{i,m'}) \underset{i=1,\dots,n \ \wedge \ j \in B_{i,m'}}{\circledast} \psi_i(\mathbf{c}_{i,j})\Big)^{-1} \approx \mathbf{s}_{i,m} \tag{6}$$

A.2 Different configuration of SHARCS

End-to-end It is possible to train all SHARCS components simultaneously, allowing a joint optimisation of the task and the concepts found. Therefore, it is also possible to include the loss of

the local tasks in Equation 3. However, to use local supervision, we need to implement inside the model n local label predictor function $f_1, \ldots, f_n \in \mathbb{N}$, one for each modality $i = 1, \ldots, n$. The local label predictor function $f_i : C_i \to Y_i$ maps the local concepts from the i-th modality to the downstream local task space $Y \subseteq R^{l_i}$, where l_i is the number of classes of the local task of the modality i. Therefore the objective function to minimise is the following:

$$\mathcal{L}(\mathbf{y}, \hat{\mathbf{y}}, \mathbf{s}) = \mathcal{T}(\mathbf{y}, \hat{\mathbf{y}}) + \frac{\lambda}{|M|} \sum_{(i,q) \in M \subseteq \binom{\{1,\ldots,n\}}{2}} \left|\left|\mathbf{s}_i - \mathbf{s}_q\right|\right|_2 + \sum_{i=1}^{n} \beta_i \mathcal{T}_i(\mathbf{y_i}, \hat{\mathbf{y}}_\mathbf{i}) \tag{7}$$

where $\beta_i \in \mathbb{R}$ is a hyperparameter that controls the strengths of the local loss $\mathcal{T}_i$.

Sequential The training process of this method is split in two parts. In the first one, a model similar to Concept Multimodal is trained. Therefore, unimodal models $g_1, \ldots, g_n$ are utilised to compute local concepts, which are concatenated and passed through the label predictor function to solve the downstream task. This part of the entire architecture is trained first, using an objective function equals to $\mathcal{T}$, solving the task using local concepts. Then, the concept encoders functions $g_1, \ldots, g_n$ are frozen. In the second part of the training, local concepts are projected into the shared space by $h_1, \ldots, h_n$, concatenated and used by f to make the final prediction. At this point, the standard loss described in Equation 3 is applied.

Local pre-training In this approach, SHARCS' single modality components $g_1, \ldots, g_n$ are trained first, using the same local label predictor functions $f_1, \ldots, f_n$ described in the end-to-end approach to make a prediction. Each is trained using their specific local loss $\mathcal{T}_i$. Then, the concept encoders functions $g_1, \ldots, g_n$ are frozen, while the other SHARCS' modules are employed and trained using the standard objective function described in Equation 3.

A.3 Concept Finding on Graph

Although our solution is model agnostic, it is important to treat every modality properly. Therefore, we slightly modify the concept encoder function when it is composed of a Graph Neural Network. Specifically, we applied a modified version of the Concept Encoder Module (CEM)[16]. In this case, the concept encoder function g_i is composed of a Graph Neural Network $\phi_i : X_i \to H_i$, a Gumbel Softmax [10] to find the "node concepts", an add pooling over the nodes of the graph, a batch scaling function and a sigmoid Function. Therefore to find $\mathbf{c}_{im}$, where i is a graph modality, the equation becomes the following:

$$\mathbf{t}_{i,m} = \phi_i(\mathbf{x}_{i,m}) \qquad \mathbf{n}_{i,m} = \sum_{d \in \mathbf{x}_{i,m}} \sigma(\mathbf{t}_{i,m,d}) \tag{8}$$

$$\mathbf{c}_{i,m} = \sigma\Big(\mathbf{n}_{i,m} \underset{j \in B_{i,m}}{\circledast} \mathbf{n}_{i,j}\Big)^{-1} \tag{9}$$

where ϕ_i represents the Graph Neural Network applied to the modality i, which outputs the representation of each node d of graph m in the modality i, σ is the Gumbel Softmax, and $\mathbf{n}$ represents the sum over the node concept of the graph m. Therefore, in our solution, the graph concept is related to the occurrences of each node concept.

The issue with CEM is that when it aggregates node concepts, there is no one-to-one mapping between a set of node concepts and graph concepts. This could lead to giving the wrong concept to a graph. Figure 3 shows an example of a situation where two different graphs end up with the same concepts.

A.4 Code, licences and Resources

Libraries For our experiments, we implemented all baselines and methods in Python 3.9 and relied upon open-source libraries such as PyTorch 2.0 [20] (BSD license), Pytorch Geometric 2.3 [5] (MIT license) and Sklearn 1.2 [21] (BSD license). In addition, we used Matplotlib [9] 3.7 (BSD license) to produce the plots shown in this paper and Dtreeviz[1] 2.2 (MIT license) to produce the

[1] `https://github.com/parrt/dtreeviz`

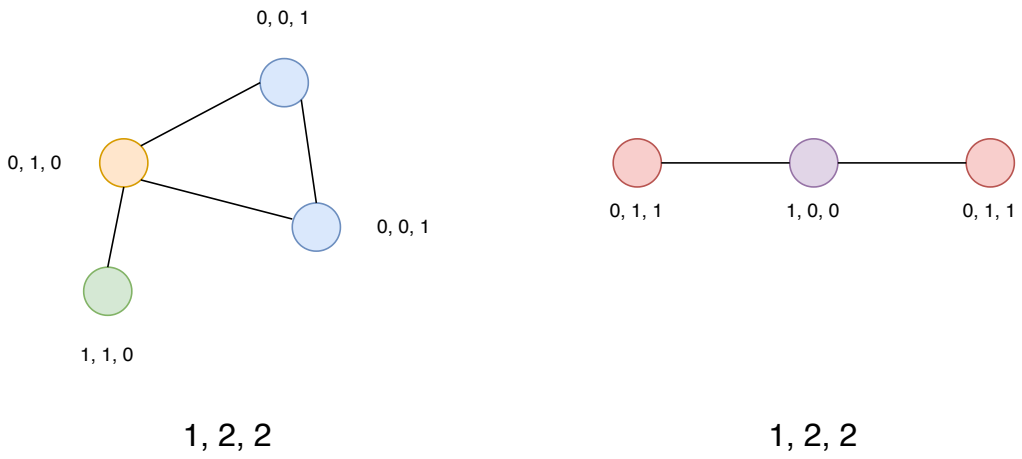


Figure 3: An example of two different graphs with a different set of node concepts described with the same graph concept.

tree visualisations. We will publicly release the code to reproduce all the experiments under an MIT license.

B Experiments details

Modeling details. As discussed earlier, SHARCS learns modality-specific concepts before combining them in a shared space. Therefore, since we consider tasks that combine different modalities, we use different models. Specifically: (i) for tabular data, we use a 2-layer Feed Forward Network; (ii) for images, a 2 layers CNN (MNIST+Superpixels, HalfMNIST) or a pre-trained ResNet18 [8] (CLEVR); (iii) for text, a 2-layer Feed Forward Network after computing the text representation with TF-IDF; and (iv) for graphs, 4 layers of GCN [12] (XOR-AND-XOR) or 2 layers of Spline CNN [6] (MNIST+Superpixels, HalfMNIST). Note that, since in this paper, we are focusing on evaluating the efficacy of SHARCS in a multimodal setting rather than pursuing state-of-the-art performance; all approaches use the same (local) backbone architectures. Nevertheless, as SHARCS is model agnostic, these can be easily extended to more sophisticated (but likely less efficient) architectures. Appendix C provides further details about model compositions and used hyperparameters in each experiment.

Baselines and experiments. We begin by examining SHARCS' multimodal capabilities. In our initial experiments, we compare SHARCS with models trained solely on single modalities. These unimodal models include both basic concept-less models and concept-based variations. In the subsequent experiments, we assess SHARCS' performance against several multimodal baseline models. These baselines consist of: (i) A standard multimodal approach called 'Simple Multimodal,' which combines uninterpretable embedded representations from individual local models (ii) A concept-based variant known as 'Concept Multimodal,' similar to the previous approach but additionally computes and uses local concepts without sharing them (iii) A 'Relative Representations' multimodal approach [18], which constructs relative mapped representations of each sample in relation to a given anchor within a shared space. This approach requires a two-stage training process: first for building representations for each modality and then for mapping them in the shared relative space. Furthermore, we consider a practical multimodal scenario involving missing modalities. In this setup, we train multimodal models using both modalities, but during inference, one of the modalities is replaced with an auxiliary one. For example, instead of representing a six as an image and a four as a graph, we represent both a six and a four as images.

Evaluation metrics. We repeat each experiment several times (three times in the case of CLEVR and five times for the other three) and report a mean and standard error for each metric we use. Each model has been evaluated using test classification accuracy to evaluate multimodal generalisation performance. Furthermore, we also report the completeness score to quantitatively assess the concept quality (for SHARCS and Concept Multimodal). The completeness score assesses how the learnt concepts are suitable to solve the downstream task. To compute it, we train a decision tree, which

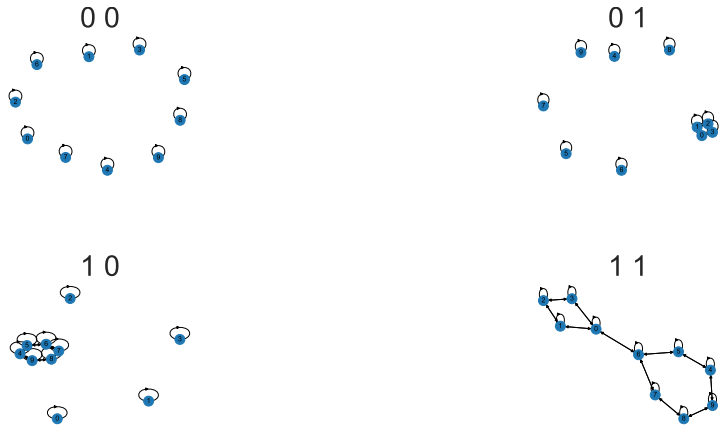


Figure 4: Examples of the conversion from the four main families of graphs to the meaningful bits of the tabular data in the XOR-AND-XOR dataset. In the dataset, they have some additional random edges.

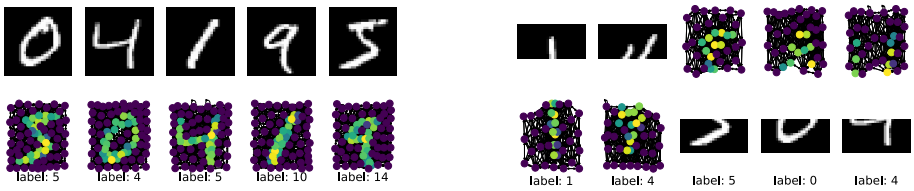
takes the binarised global concepts at the input. To evaluate the performance of SHARCS to the ones of the 'Relative representation' and 'Concept Multimodal' variants in the missing modality settings, we compute their accuracy in this scenario.

Tasks and datasets. We evaluate our hypotheses on four multimodal tasks, each leveraging a pair of multimodal datasets such as tabular, image, graph, and text data. The four multimodal, or global, tasks are designed such that the models need to leverage both modalities in order to provide correct predictions. Models that will learn only from one of the modalities will be able to solve a partial (local) single-modality task but will typically exhibit random performance on the global multimodal task. Furthermore, we test the interpretability of SHARCS and its ability to cope with real-world scenarios when a modality is missing.

The first task, *XOR-AND-XOR*, considers multimodal settings with tabular and graph data, each modelling a local/partial XOR task. The entire dataset contains 1000 samples for each modality. The tabular modality consists of bit-strings (2 used for solving the 'xor' and 4 random), while the graph modality comprises 4 types of graphs (the label is binarized and used for solving the 'xor' task, as Figure 4 shows). The global multimodal task is an 'and' binary problem, combining the outcome of the two local 'xor' tasks. The second task is *MNIST+Superpixels*, comprised of 60000 pairs of image modality (MNIST [3]) and a graph modality, the latter representing a superpixel-graph of an MNIST image [17]. While the local tasks are treated as classical classification tasks (from an image and a graph, respectively), the global multimodal task concerns predicting the sum of the two digits. Figure 5a shows five samples from this dataset, including the global label. Next, we consider *HalfMNIST*, which combines 60000 samples of an image and a graph modality. Here the task is to perform (MNIST) classification, but each modality comprises one part of the sample (the top/bottom half of an image or graph). Figure 5b shows five samples from this dataset. Finally, the last task builds on *CLEVR*, a standard benchmark in visual question answering comprised of image and text modalities. Specifically, in our multimodal setting, we follow [25] and produce our own CLEVR sample dataset with 8000 samples, where instead of having a question, we generate text captions for the generated images. In turn, the multimodal task is a binary problem, predicting whether the caption matches the image. Figure 6 shows five samples taken from this dataset, the top row represents the captions, while the bottom is about the images.

C Models details

In this section, we describe in detail the configuration of SHARCS used in each experiment. Then, we add only the missing or different information needed to build the other models used, as most of the details are in common between our solutions and baselines.



(a) (b)

Figure 5: (a) Examples from the MNIST+Superpixels dataset. The shown label is related to the task, which is the sum of the two digits. (b) Examples from the HalfMNIST dataset. The shown label is related to the task, which is the digit represented by joining both parts. Each half can be represented with one of the two modalities.

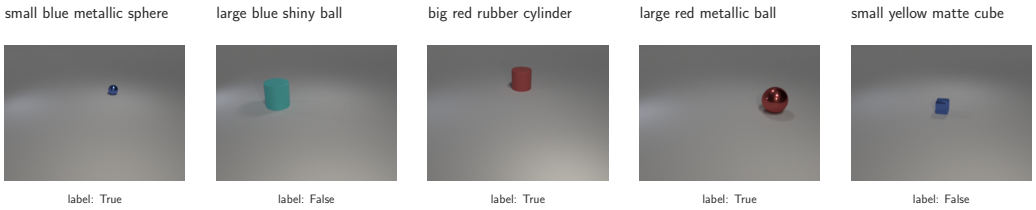


Figure 6: Examples from the CLEVR dataset, where there is a text caption and an image of an object. The label is True if the caption correctly describes the image, otherwise is False.

In general, single modality models used only the DL model inside of the respective g_i, with (or without) a sigmoid function, if it is a concept-based (concept-less) solution. Simple Multimodal and Relative representation solutions employ the DL models inside g_i and the label predictor f, while Concept Multimodal also uses batch scaling and the sigmoid inside g_i.

C.1 XOR-AND-XOR

On this task, we trained SHARCS with the end-to-end configuration, as we do not have local supervision. It is composed of two g_i concept encoder functions, one for each modality. To handle the graph modality, the DL model inside of g_1 is composed of 5 layers of Graph Convolutional Networks [12] with LeakyReLU as the activation function. The input size is 1, the hidden size of all the intermediate layers is 30, while the output dimension of g_1 is 7. On the other hand, a simple 2-layer MLP with a ReLU as the activation function is the DL model of g_2, which takes tabular data as input. The input size is 8, the hidden size is 30, and the output dimension is equal to 7. SHARCS uses Batch Normalisation as batch scaling and Sigmoid to compute concepts, but on the graph modality follows the approach described in Appendix A.3. The second set of concept encoders h_1 and h_2 are 2-layer MLPs with a ReLU as the activation function, with an input dimension of 8, as well as the hidden and output size. Finally, the label prediction function f is a 2-layer MLP with a ReLU as the activation function, with an input dimension of 16, a hidden size of 10 and an output dimension equals to the number of classes, which is 2.

An additional detail for single modality models is their label prediction function f_i, one for each modality, which is a 2-layers MLPs with a ReLU as the activation function, with an input dimension of 8, a hidden size of 10 and an output dimension of 2.

In terms of learning process, we used a Binary Cross Entropy Loss (BCELoss) with Logits (which incorporates a sigmoid layer before computing the BCELoss) as $\mathcal{T}$, a λ equals to 0.1, and at every iteration, we took 10% of randomly draw samples to compute the distance. Other hyperparameters used to train the models are the Batch Size used (64), the number of epochs (150) and the Learning Rate used by an Adam optimizer (0.001). However, we train the unimodality models of Relative representation models for 150 epochs and its label predictor function for other 150 epochs.

C.2 MNIST+Superpixels and HalfMNIST

On MNIST+Superpixels and HalfMNIST, we used an almost identical setup. We trained SHARCS with the local pre-training configuration, as we have local supervision. It is composed of two g_i concept encoder functions, one for each modality. To handle the graph modality, the DL model inside of g_1 is composed of 2 layers of SplineCNN [6] with ELU as the activation function, similar to the SplineCNN model described in the original paper. Therefore, a max pooling operator based on the Graclus method [4] is applied after every layer. The input size is 1, the hidden size of all the intermediate layers is 32, and the output dimension of g_1 is 12. On the other hand, a Convolutional Neural Network is the DL model of g_2. It is composed of the following layers: a Convolutional Layer (input channel=1, output channel=16, kernel size=5, padding=2, stride=1), a ReLU, a MaxPool with a kernel size of 2, a Convolutional Layer (input channel=16, output channel=16, kernel size=5, padding=2, stride=1), a ReLU, a MaxPool with a kernel size of 2, then the output is flattened and taken as input from a 2-layer MLP with a ReLU as the activation function, with a hidden dimension of 64 and output size of 12. Moreover, SHARCS uses Batch Normalisation as batch scaling and sigmoid to compute concepts, but on the graph modality follows the approach described in Appendix A.3. The second set of concept encoders h_1 and h_2 are 2-layer MLPs with a ReLU as the activation function, with an input dimension of 12, as well as the output size and a hidden size of 64. Finally, the label prediction function f is a 2-layer MLP with a ReLU as the activation function, with an input dimension of 24, a hidden size of 128 and an output dimension equals to the number of classes, which is 19 for MNIST+Superpixels and 10 for HalfMNIST. As we apply the local pre-training configuration, in the first part of the training, we used some local label predictor function f_i, one for each modality. They are 2-layer MLPs with a ReLU as the activation function, with an input dimension of 12, a hidden size of 64 and an output dimension equals to the number of classes of the local task, which is 10 for both datasets. Other unimodal baselines also use these local label predictor functions.

Regarding the learning process, we used a BCELoss with Logits both with local and global tasks, a λ equals to 0.1, and at every iteration, we took 10% of randomly drawn samples to compute the distance. Other hyperparameters used to train the models are the Batch Size used (64), the number of epochs used to pretrain the unimodal models (15) and the additional epochs used to train the second part of SHARCS (15). The learning rate used by the Adam optimiser is equal to 0.01 for the Graph Neural Network and 0.001 for all the other layers of the model. However, we train the unimodality models of Relative representation models for 15 epochs and its label predictor function for other 20 epochs.

C.3 CLEVR

On this task, we trained SHARCS with the sequential configuration, as we do not have local supervision and want to discover local concepts that are not influenced by the other modality. It is composed of two g_i concept encoder functions, one for each modality. To handle the image modality, the DL model inside of g_1 is a pretreated ResNet18 [8], followed by a Dense layer that reduced the output size of the ResNet to 24. On the other hand, a simple 2-layer MLP with a ReLU as the activation function is the DL model of g_2, which takes the TF-IDF representation of the caption received as input. The input size is 22, the hidden size is 48, and the output dimension is equal to 24. SHARCS uses Batch Normalisation as batch scaling and sigmoid to compute concepts. The second set of concept encoders h_1 and h_2 are 2-layer MLPs with a ReLU as the activation function, with an input dimension of 24, as well as the hidden and output size. Finally, the label prediction function f is a 2-layer MLP with a ReLU as the activation function, with an input dimension of 48, a hidden size of 10 and an output dimension equals to the number of classes, which is 2.

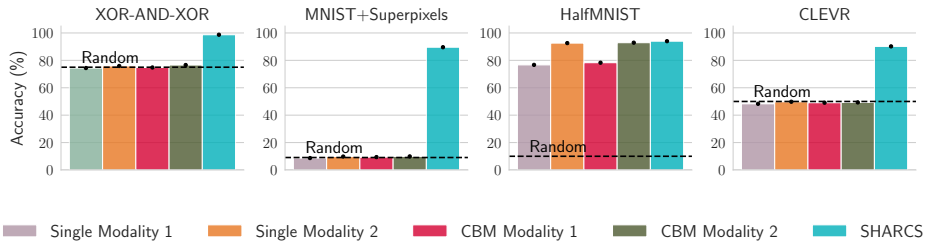


Figure 7: Accuracy of unimodal models and SHARCS on all datasets. SHARCS outperforms all the other models on all tasks.

Table 2: Accuracy (%) and Completeness Score (%) of SHARCS compared to non-interpretable unimodal models (Simple Modality 1 and Simple Modality 2), non-interpretable multimodal models (Simple Multimodal and Relative representation), interpretable unimodal models (CBM Modality 1 and CBM Modality 2) and interpretable multimodal baselines (Concept Multimodal). Generally, SHARCS achieves better (or comparable) performance than the other baselines, producing better and more compact concepts.

Model	**XOR-AND-XOR**		**MNIST+SuperP.**		**HalfMNIST**		**CLEVR**	
	Acc.	Compl.	Acc.	Compl.	Acc.	Compl.	Acc.	Compl.
Mod 1	74.4 ± 0.7	-	8.7 ± 0.1	-	76.7 ± 0.2	-	48.3 ± 0.3	-
Mod 2	75.9 ± 1.4	-	9.8 ± 0.1	-	92.6 ± 0.2	-	49.8 ± 0.1	-
CBM 1	74.8 ± 0.0	-	9.4 ± 0.1	-	78.3 ± 0.1	-	49.1 ± 0.5	-
CBM 2	76.6 ± 1.3	-	9.9 ± 0.2	-	92.9 ± 0.1	-	49.3 ± 0.4	-
Simple	99.3 ± 0.5	-	86.6 ± 3.0	-	94.2 ± 0.2	-	59.5 ± 9.5	-
Concept	99.0 ± 0.8	96.2 ± 1.2	88.2 ± 0.1	78.9 ± 1.4	93.9 ± 0.0	91.3 ± 0.1	90.1 ± 1.0	**82.3** ± 1.2
Relative	**99.5** ± 0.3	-	80.4 ± 0.2	-	**95.6** ± 0.1	-	48.7 ± 0.5	-
SHARCS	98.7 ± 0.5	**98.0** ± 1.2	**89.6** ± 0.1	**88.7** ± 0.2	94.0 ± 0.1	**92.6** ± 0.3	**90.2** ± 0.2	81.5 ± 1.1

An additional detail for single modality models is their label prediction function f_i, one for each modality, which is a 2-layers MLPs with a ReLU as the activation function, with an input dimension of 24, a hidden size of 24 and an output dimension of 2.

In terms of learning process, we used a BCELoss with Logits, a λ equals to 0.1, and at every iteration, we took the samples with the label equals to True out of 20% of randomly drawn samples to compute the distance. Other hyperparameters used to train the models are the Batch Size used (64), the number of epochs used by all models and in the first part of the training of SHARCS (30), the additional epochs used in the second part of the training of SHARCS (20) and the Learning Rate used by an Adam optimizer (0.001). In addition, we train the unimodality models of Relative representation models for 30 epochs and its label predictor function for other 20 epochs.

D Additional results

This section includes additional results and consideration of the experiments presented in Section 3.

Broader Impacts We do not believe this approach can have a direct harmful impact when applied in AI systems. On the contrary, it can positively influence the development of models for safety-critical domains, such as healthcare.

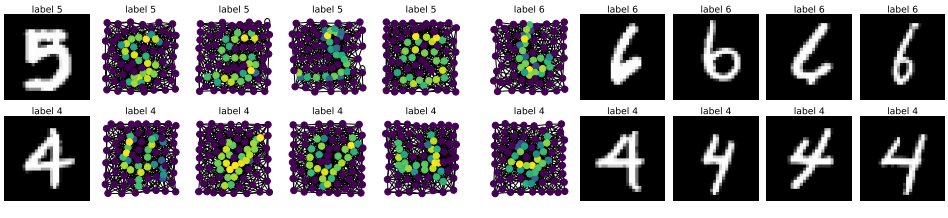
Detailed results of experiments Figure 7 shows the performance of unimodal models compared to SHARCS in all tasks. It is clear how 3 out of the 4 datasets we designed are not solvable by unimodal models, proving our design choice. Furthermore, Table 2 shows the Accuracy for all the models trained and the Completeness Score for the multimodal interpretable models. It gives more detailed results and compares together all the trained models. On the other hand, Table 3, shows the result of an analysis we performed on CLEVR, where we checked for each model which characteristics of the retrieved sam- ple matched with the ones of the object used as the source.

Interpretability We present the visual results for some of the dataset to give a better idea of the performance of our solution. We show the retrieved examples per modality, the learnt shared space

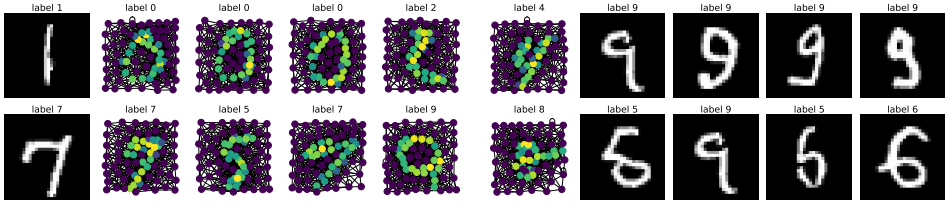
Table 3: Accuracy (%) of Relative representation, Concept Multimodal and SHARCS in retrieving a specific characteristic in a modality using the other. SHARCS attains higher figures than other models on every characteristic.

Model	Modality	Shape	Size	Material	Color	Mean
Concept	Text	31.7 ± 2.3	46.0 ± 3.0	52.1 ± 0.2	16.2 ± 3.7	36.5 ± 0.6
	Image	30.0 ± 0.2	45.4 ± 5.3	51.3 ± 2.6	10.8 ± 1.3	34.3 ± 0.6
Relative	Text	29.9 ± 1.0	50.5 ± 0.7	50.0 ± 0.6	13.3 ± 1.2	35.9 ± 0.3
	Image	33.0 ± 1.0	49.6 ± 0.2	49.0 ± 1.0	11.1 ± 0.7	35.6 ± 0.2
SHARCS	Text	$\mathbf{56.8} \pm 1.4$	$\mathbf{63.6} \pm 3.5$	$\mathbf{53.9} \pm 1.8$	$\mathbf{30.2} \pm 5.6$	$\mathbf{51.1} \pm 2.0$
	Image	$\mathbf{51.4} \pm 2.4$	$\mathbf{61.5} \pm 1.7$	$\mathbf{53.4} \pm 2.6$	$\mathbf{27.5} \pm 4.0$	$\mathbf{48.5} \pm 1.9$

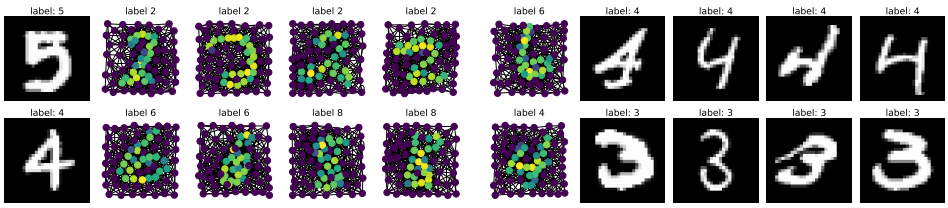
and the decision tree. Figure 8 shows the retrieved examples by SHARCS, Relative Representation and Concept Multimodal in the MNIST+Superpixels dataset. Finally, Figure 9 shows the retrieval capability of these models on the CLEVR dataset. In all these experiments, it can be seen that the quality of the retrieved examples is higher than the others, where the Relative Representation is not always accurate, and the Concept Multimodal resembles random retrieval. The second set of images visually confronts the shared space learnt by SHARCS and Concept Multimodal. For this purpose, we visualise the tSNE representation of the shared concepts for SHARCS and the local concepts for Concept Multimodal. Figure 10 shows these shared spaces for the MNIST+Superpixels dataset and Figure 11 for CLEVR. It is clear how the concept representation learnt by SHARCS for one modality overlaps with that for the other, especially when considering semantically similar examples from different modalities that are closer in the space representation. All these results are expected by design since we force the model to produce the shared space with these properties. Finally, part of the decision trees used to compute the completeness score is visualised. At every split, it shows the concept that is considered to make the decision, and it can be active (right branch) or non-active (left branch). If the node is not the roof, it also shows three samples with the highest (concept active) or the lowest (concept non-active) value for the concepts of the previous split, among the ones that respect all the previous split conditions. Each leaf shows the class distribution of the samples that it represents, in addition to the most characteristic samples. Moreover, the root of the tree uses the most influential concept for the classification task, as it is by definition the one that brings the highest Information Gain, and the same is applicable to the following splits. For example, Figure 12 shows the decision tree used in the XOR-AND-XOR dataset. Specifically, you can see that if Concept 11 is active, the prediction is always Class 0 (False). As you can see, if a sample has Concept 11 active, it means that it has both tabular significant digits equal to 1, which implies that the local XOR operation is False and as a consequence the global AND operation is False, no matter what is the other modality. Furthermore, the following split is focused on Concept 4, which is curiously the corresponding concept in the shared space of the graph modality for Concept 11 (7 is the number of concepts per modality, so 11 - 7 = 4). This split represents the same underlying idea as the previous one but for the graph modality. If the concept is active, it means that the graph is connected (False in the local XOR operation). Therefore, it shows also how the concepts from one modality are related and translated into the other, confirming that the concept shared space created is meaningful. Finally, Figure 13 shows part of the first three layers of the Decision tree used in CLEVR.



(a)

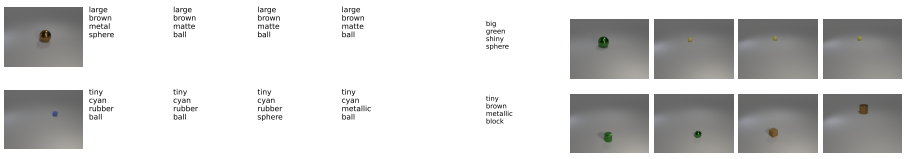


(b)

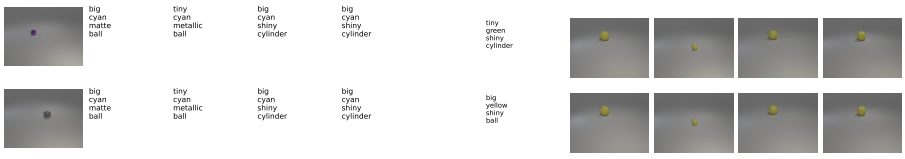


(c)

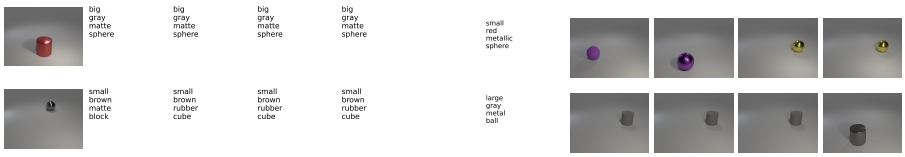
Figure 8: Retrieval examples obtained by (a) SHARCS, (b) Relative representation, and (c) Concept Multimodal on the MNIST+Superpixels dataset. The top two rows are samples of retrieved graphs using images, while the bottom two are retrieved images using graph samples.



(a)

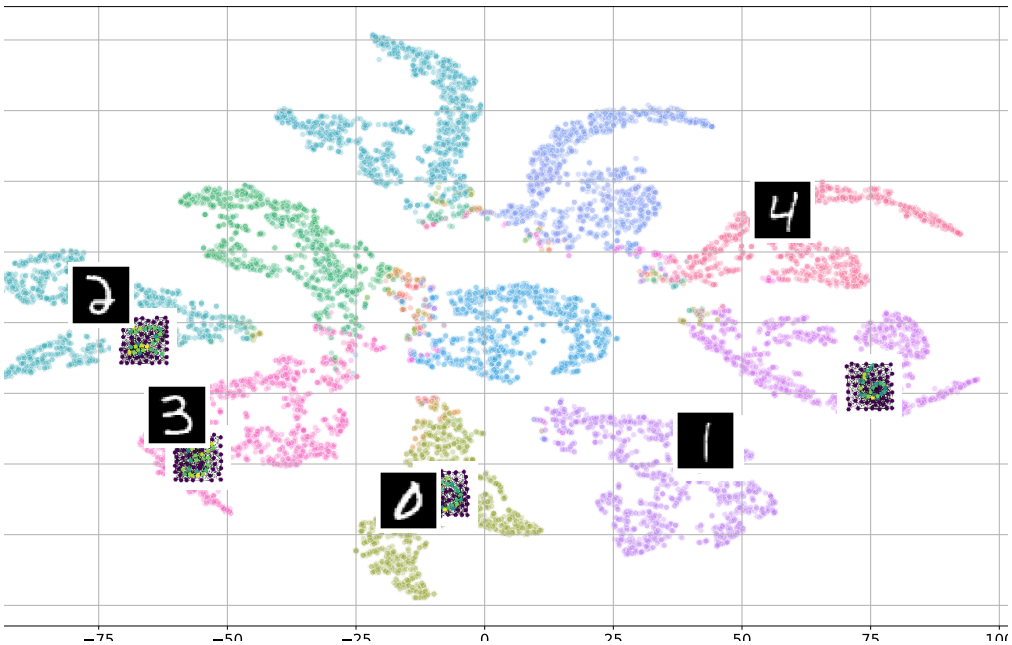


(b)

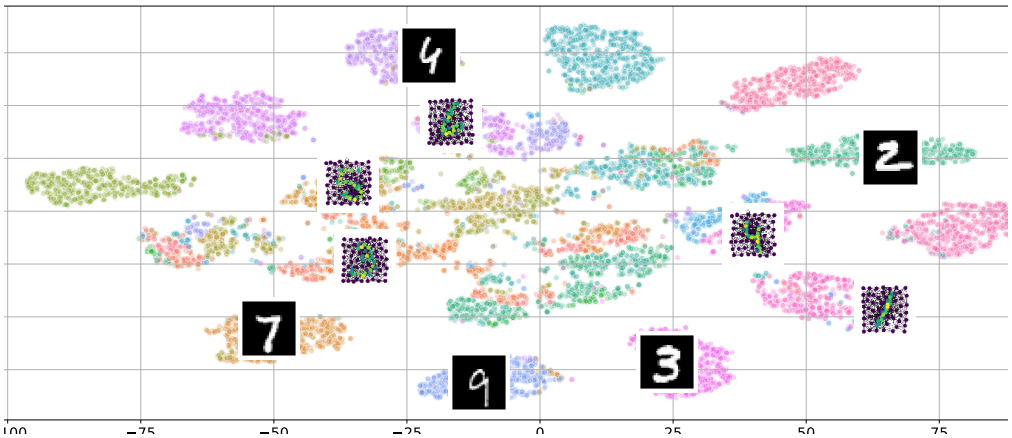


(c)

Figure 9: Retrieval examples obtained by (a) SHARCS, (b) Relative representation, and (c) Concept Multimodal on the CLEVR dataset. The top two rows are samples of retrieved text using images, while the bottom two are retrieved images using graph samples.

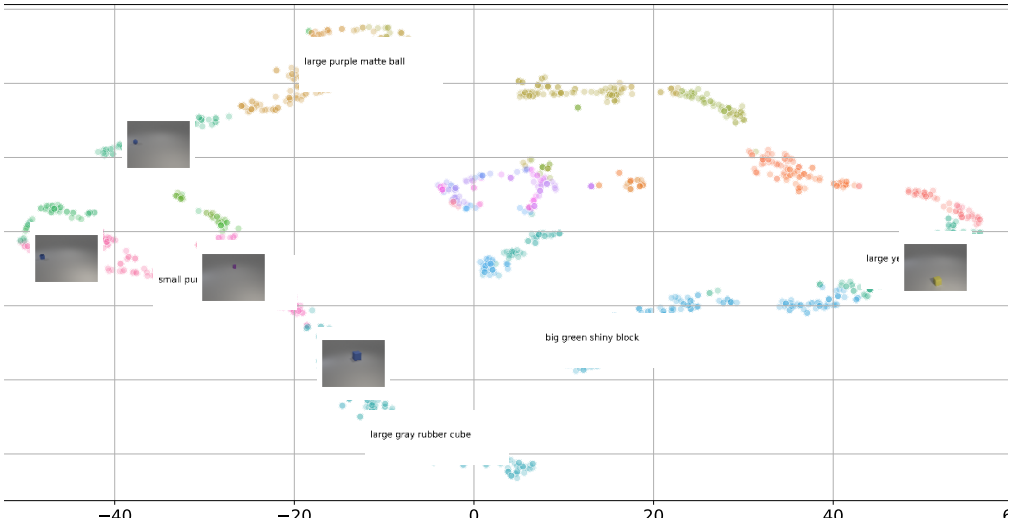


(a)

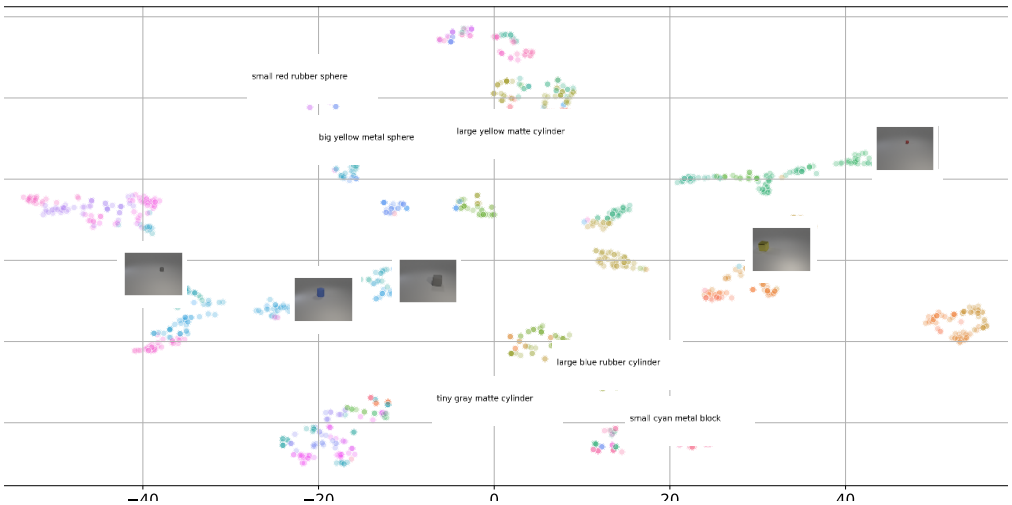


(b)

Figure 10: tSNE plot of the concept space. The images represent the centroid of the top-5 common concepts per modality in the MNIST+Superpixels dataset (a) SHARCS (b) Concept Multimodal



(a)



(b)

Figure 11: tSNE plot of the concept space. The images represent the centroid of the top-5 common concepts per modality in the CLEVR dataset (a) SHARCS (b) Concept Multimodal

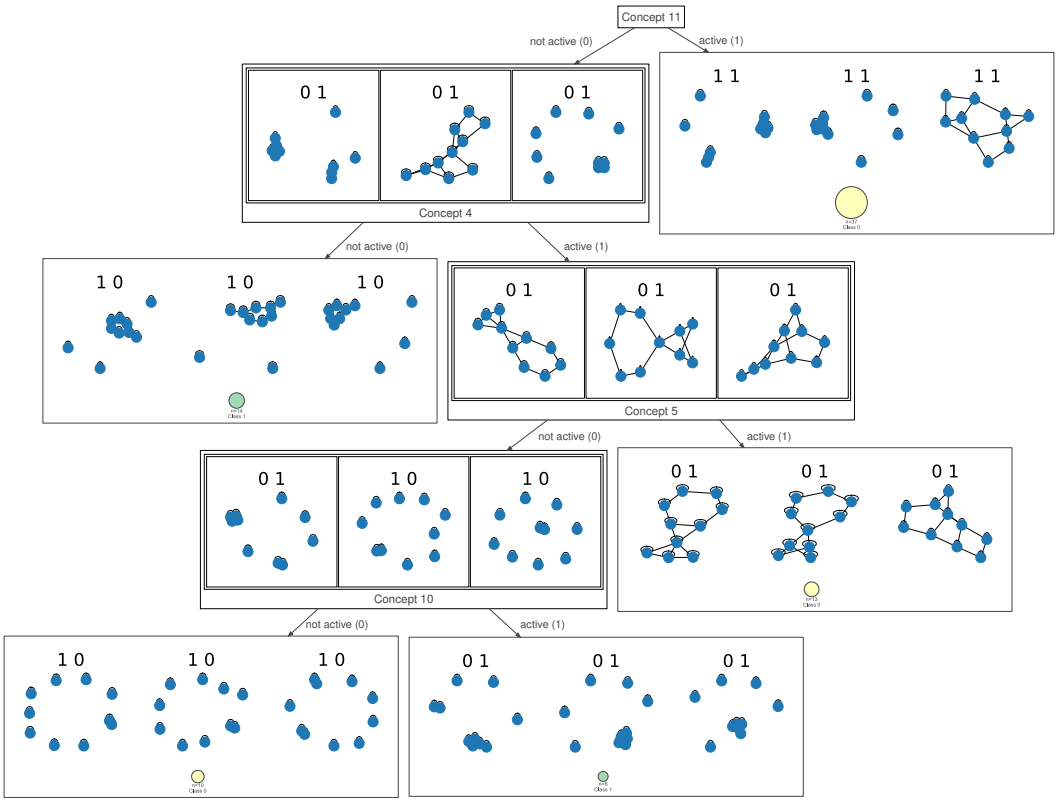


Figure 12: Decision tree visualisation of SHARCS concepts on the XOR-AND-XOR dataset. Every split shows the combined concept closer to the cluster's centroid lower and greater than the splitting criteria. In addition, each leaf shows the class distribution of the samples that it represents.

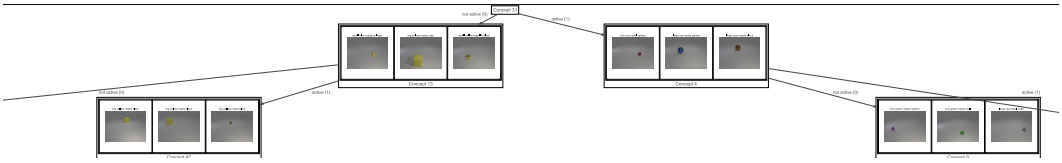


Figure 13: Visaulisation of part of the first 3 layers of a Decision tree trained on SHARCS concepts on the CLEVR dataset. Every split shows the combined concept closer to the cluster's centroid lower and greater than the splitting criteria. In addition, each leaf shows the class distribution of the samples that it represents.