MolVision: Molecular Property Prediction with Vision Language Models

Deepan Adak[1], Yogesh Singh Rawat[2], Shruti Vyas[2],
[1]NIT Kurukshetra, [2]Institute of AI, University of Central Florida

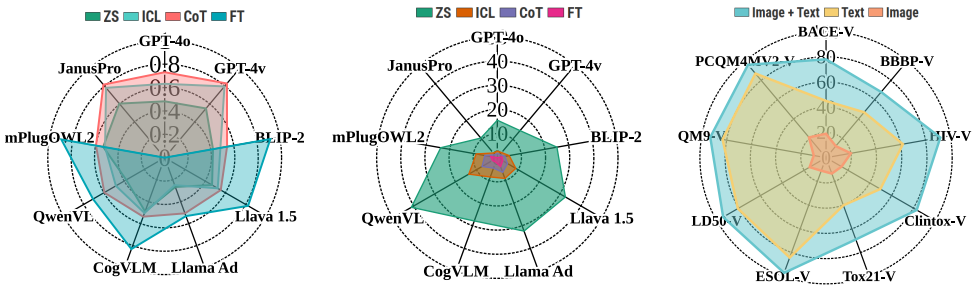

Figure 1: ***MolVision overview:*** Average performance comparison of models in zero-shot (ZS), in-context (ICL), chain-of-thoughts (CoT), and finetuning (FT) for classification (***Left*** ↑) and regression tasks (***Center*** ↓). (***Right:)*** Impact of using visual information on model performance (↑) (JanusPro).

Abstract

Molecular property prediction is a fundamental task in computational chemistry with critical applications in drug discovery and materials science. While recent works have explored Large Language Models (LLMs) for this task, they primarily rely on textual molecular representations such as SMILES/SELFIES, which can be ambiguous and structurally less informative. In this work, we introduce MolVision, a novel approach that leverages Vision-Language Models (VLMs) by integrating both molecular structure as images and textual descriptions to enhance property prediction. We construct a benchmark spanning ten diverse datasets, covering classification, regression and description tasks. Evaluating nine different VLMs in zero-shot, few-shot, and fine-tuned settings, we find that visual information improves prediction performance, particularly when combined with efficient fine-tuning strategies such as LoRA. Our results reveal that while visual information alone is insufficient, multimodal fusion significantly enhances generalization across molecular properties. Adaptation of vision encoder for molecular images in conjunction with LoRA further improves the performance. The code and data is available at : https://molvision.github.io/MolVision/.

1 Introduction

Recent advancements in Large Language Models (LLMs) have revolutionized natural language understanding across multiple domains (1). Vision-Language Models (VLMs) extend these capabilities by integrating visual and textual modalities (2), enabling tasks such as image captioning and visual question answering. While VLMs have been extensively explored in computer vision applications, their potential in scientific domains—particularly chemistry—remains largely unexplored.

Recent works such as ChemLLM (3) have explored LLMs for molecular property prediction using textual representations like SMILES. However, these representations have notable limitations, including non-uniqueness and syntactic instability, where structurally identical molecules may have

39th Conference on Neural Information Processing Systems (NeurIPS 2025) Workshop: 2nd Workshop on Multi-modal Foundation Models and Large Language Models for Life Sciences.

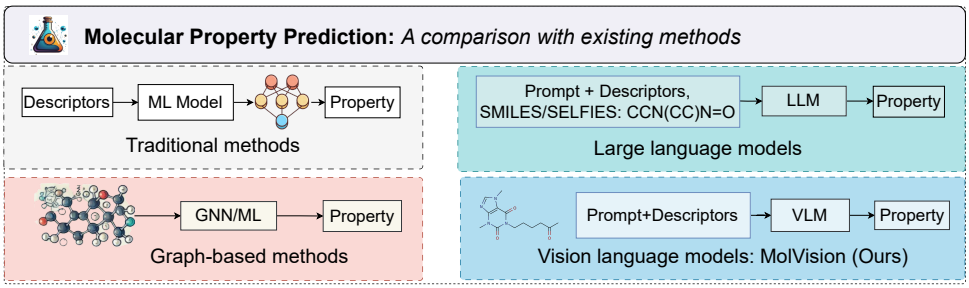

Figure 2: ***MolVision comparison:*** Comparison of relevant molecular property prediction approaches.

different textual encodings. This ambiguity introduces challenges for LLMs processing molecular structures as linear strings.

Incorporating visual information has the potential to significantly enhance molecular property prediction. Chemists typically analyze molecular structures using bond-line diagrams to infer properties such as reactivity and solubility. These visual representations encode structural information that textual descriptors may fail to capture. By leveraging VLMs, we aim to bridge this gap and improve predictive modeling in cheminformatics (Figure 2).

We introduce MolVision, a multimodal benchmark for molecular property prediction that integrates both textual and visual representations. Our benchmark spans ten diverse datasets covering classification, regression, and description tasks. We evaluate nine different VLMs and propose a contrastive strategy to adapt VLMs for molecular structures.

2 Visual Language Models for Property Prediction

We propose using visual information (molecular images) alongside textual descriptions to improve property prediction. A vision-language model processes both image and text prompt to generate textual output. The input image is processed using a vision encoder, the textual prompt through a text encoder, and features are fused via multi-modal learning for text generation.

Model variants: We study three setups: (1) *Zero-shot*: evaluation without fine-tuning or examples, (2) *Few-shot*: using in-context learning (ICL) with examples selected via Tanimoto similarity and chain-of-thought (CoT) prompting, and (3) *Fine-tuning*: using LoRA adaptation while keeping vision and text encoders frozen. We evaluate nine VLMs: BLIP-2 (4), LLaVA 1.5 (5), LLaMA Adapter V2 (6), CogVLM (7), Qwen-VL (8), mPLUGOWL2 (9), Janus-Pro 7B (10), GPT-4V, and GPT-4o.

3 MolVision Benchmark

Our benchmark includes ten datasets covering diverse molecular properties. We augment existing property prediction datasets with skeletal images generated using RDKit (11). For classification tasks, we frame questions as True/False predictions. For regression, models generate numerical values, and for description tasks, textual outputs. The datasets include: BACE-V (bioactivity) (12), BBBP-V (blood-brain barrier penetration) (13), HIV-V (HIV activity) (14), ClinTox-V (toxicity) (12), Tox21-V (toxicity) (15), ESOL-V (solubility) (16), LD50-V (acute toxicity) (17), QM9-V (quantum properties) (18), PCQM4Mv2-V (HOMO-LUMO gap) (19), and ChEBI-V (molecular descriptions) (20).

4 Experiments and Results

Few-shot Performance: Table 1 shows few-shot ICL performance for classification tasks, while Table 2 presents regression results. All models show improvements over zero-shot evaluation. GPT-4o and GPT-4v achieve the best performance across both classification and regression tasks, followed by Janus-Pro 7B among open-source models.

Impact of Visual Information: Table 3 compares performance using text-only, image-only, and combined inputs. Results show that image-only inputs are insufficient, but combining visual and textual data significantly improves performance.

Table 1: **Few-shot classification performance:** Accuracy (F1-score) using ICL with k=2.

Models	BACE-V ↑	BBBP-V↑	HIV-V ↑	ClinTox-V ↑	Tox21-V ↑	Average ↑
GNN Models						
UniMol (21) †	0.78(0.67)	0.82(0.70)	0.82(0.73)	0.94(0.83)	0.77(0.65)	0.83(0.72)
Molca (22) †	0.79(0.73)	0.74(0.72)	0.89(0.84)	0.93(0.84)	0.80(0.72)	0.83(0.77)
VLM [ICL k=2]						
GPT-4o	0.56(0.53)	**0.77(0.81)**	0.82(**0.56**)	0.59(0.44)	0.42(0.58)	0.63(0.58)
GPT-4v	0.72(0.66)	0.63(0.60)	**0.95**(0.44)	**0.96**(**0.94**)	0.72(0.52)	**0.80(0.63)**
Janus Pro 7B	**0.78(0.71)**	0.68(0.62)	0.92(0.52)	0.83(0.56)	0.69(0.49)	0.78(0.58)
BLIP-2	0.36(0.52)	0.37(0.29)	0.60(0.30)	0.34(0.36)	0.75(0.42)	0.48(0.38)
Llava 1.5 13B	0.49(0.48)	0.44(0.39)	0.24(0.34)	0.64(0.76)	**0.81**(0.31)	0.52(0.46)
Llama Ad v2 7B	0.28(0.29)	0.18(0.11)	0.19(0.17)	0.29(0.12)	0.31(0.21)	0.25(0.18)
CogVLM	0.48(0.51)	0.40(0.37)	0.31(0.21)	0.64(0.62)	0.69(**0.65**)	0.50(0.47)
QwenVL	0.69(0.46)	0.30(0.12)	0.28(0.36)	0.52(0.48)	0.62(0.63)	0.48(0.41)
mPlugowl2	0.59(0.32)	0.35(0.38)	0.62(0.29)	0.34(0.42)	0.69(0.56)	0.52(0.39)

Table 2: ***Few-shot performance for regression:*** A comparison of error in prediction using in-context learning (ICL k=2) and chain-of-thoughts (CoT) with traditional and LLM based approaches.

Model	ESOL-V ↓		LD50-V ↓		QM9-V ↓		PCQM4M-V ↓		Average ↓	
Traditional approaches										
GenRA(23)	-		0.58		-		-		-	
Unimol (21)	0.788		-		0.00467		0.070		-	
Vision-Language Models	ICL	CoT	ICL	CoT	ICL	CoT	ICL	CoT	ICL	CoT
GPT-4o	0.98	0.77	0.87	0.60	**8.38**	5.24	0.68	0.53	2.73	1.78
GPT-4v	0.99	**0.71**	**0.71**	0.59	8.62	4.66	0.77	0.66	2.78	1.66
Janus-Pro 7B	**0.61**	0.89	0.72	0.60	8.53	**4.42**	**0.62**	**0.38**	**2.52**	**1.57**
BLIP-2	1.99	1.07	0.73	**0.49**	16.01	10.09	1.30	1.25	5.01	3.23
Llava 1.5 13B	6.01	2.18	0.94	0.69	27.00	15.21	1.42	1.49	8.84	4.89
Llama Ad v2 7B	3.08	2.17	3.36	2.12	28.09	19.24	4.06	2.36	9.15	6.47
CogVLM	1.26	1.21	3.47	0.78	25.85	15.15	1.44	1.24	8.50	4.59
Qwen VL	3.96	2.89	1.06	0.63	38.92	18.08	10.61	9.56	13.64	7.29
mPlugOWL2	1.46	1.50	0.94	0.71	29.33	19.17	1.84	1.62	8.89	5.25

Table 3: ***Impact of visual information:*** A performance comparison showing the impact of visual information (molecular image) when used with textual description (SMILES). Accuracy is shown for classification (BACE-V, BBBP-V, HIV-V, Clintox-V (CV), Tox21-V (TV)), and MAE (LD50-V, QM9-V and PCQM4Mv2-V (PV)) and RMSE (ESOL-V) is shown for regression tasks.

Model	Input	BACE-V ↑	BBBP-V ↑	HIV-V ↑	CV ↑	TV ↑	ESOL-V ↓	LD50-V ↓	QM9-V ↓	PV ↓
BLIP-2	Text Only	0.71	0.76	0.69	0.64	0.78	9.89	7.80	31.76	11.31
	Image Only	0.15	0.09	0.10	0.13	0.18	32.16	31.23	149.12	36.96
	Image+Text	0.86	0.93	0.92	0.89	0.99	1.07	0.49	4.92	1.99

Table 4: ***Performance comparison for proposed contrastive learning:*** Evaluation of different contrastive learning approaches across multiple molecular datasets.

Method	BACE-V Acc(F1)	BBBP-V Acc(F1)	HIV-V Acc(F1)	Clintox-V Acc(F1)	Tox21-V Acc(F1)	ESOL-V (RMSE)	LD50-V (MAE)	QM9-V (MAE)	PCQM-V	Chebi-V (Average)
LoRA	0.86(0.83)	0.93(0.96)	0.92(0.76)	0.89(0.93)	0.99(0.80)	1.76	0.78	4.92	0.24	58.59
Aug	0.87(0.85)	0.94(0.95)	0.93(0.78)	0.90(0.94)	0.97(0.83)	0.90	0.20	4.09	0.21	60.98
T-Aug	**0.91(0.88)**	**0.95(0.96)**	**0.95(0.84)**	**0.93(0.93)**	**0.98(0.89)**	**0.58**	**0.10**	**2.95**	**0.12**	**63.73**

5 Contrastive Learning for Vision Adaptation

To enhance VLMs' visual representation capability for molecular structures, we analyzed vision embeddings using t-SNE and found them poorly clustered—likely due to pretraining on natural images. We fine-tuned the vision encoder using contrastive learning with NT-Xent loss, using two strategies: (1) augmented views of the same molecule (Aug), and (2) structurally similar molecules via Tanimoto similarity >0.85 (T-Aug). Table 4 shows that the similarity-based approach leads to more distinct embeddings and significantly improves performance—reducing ESOL RMSE by 35%, LD50 MAE by 51%, and boosting classification accuracy by 2-4%.

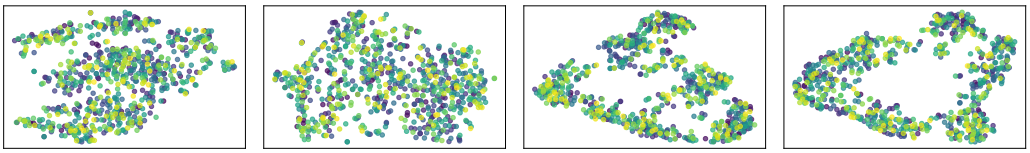

Figure 3: ***Analyzing visual features:*** The left two plots show t-SNE visualizations of visual encodings of BLIP-2 before and after cross-modal fusion respectively. The right two plots show corresponding t-SNE plots with the proposed contrastive loss using Tanimoto augmentation (T-Aug).

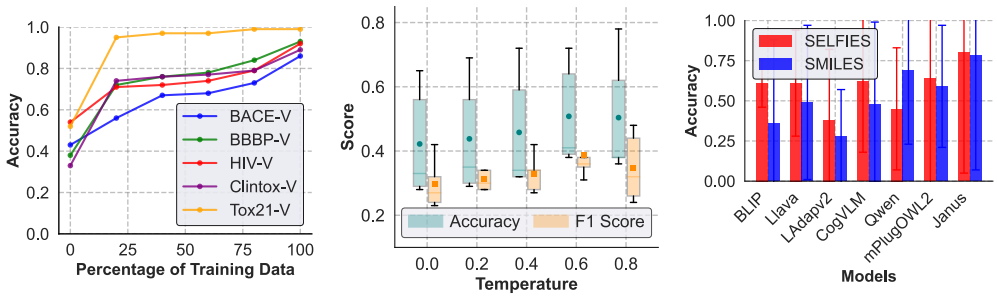

Figure 4: ***Analysis on finetuning, temperature and SELFIES vs. SMILES:*** The first plot shows the impact of percentage of finetuning data. The middle figure shows performance variation with temperature across datasets for BLIP2. The last figure shows analysis of SMILE vs SELFIES string for ICL k=2 across various models.

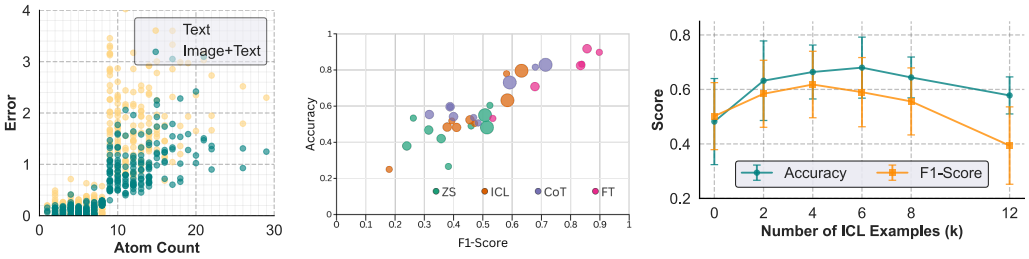

Figure 5: **Key analysis insights:** (Left) Molecular size impact on regression error showing visual data benefits for larger molecules. (Center) Model performance vs size across evaluation settings (bubble size = parameters). (Right) ICL example count effects on GPT-4o performance.

6 Discussion and Analysis

Our analysis reveals several key insights: Fine-tuning dramatically improves VLM performance with gains plateauing around 20-30% of training data, while optimal temperature settings fall between 0.3-0.7. SELFIES consistently outperforms SMILES across models in few-shot scenarios, likely due to more extensive pre-training exposure. Visual representations provide increasing benefits for larger molecules where textual encodings may lose structural information, larger models excel in few-shot settings while smaller models achieve competitive performance through fine-tuning, and ICL performance saturates beyond 4-6 examples (Figure 4, 5). These findings highlight the complementary nature of visual and textual molecular representations, with multimodal approaches particularly advantageous for complex structures where traditional string-based encodings are insufficient. VLMs struggle in zero-shot settings but improve significantly with in-context learning and targeted fine-tuning.

7 Conclusion

We present MolVision, a multimodal approach for molecular property prediction using vision-language models. Through evaluations across diverse datasets, we demonstrate the benefits of combining visual and textual molecular representations. Our proposed contrastive learning strategy using Tanimoto similarity significantly improves VLM adaptation to molecular structures. This work establishes a strong benchmark for future research exploring multimodal approaches in cheminformatics and highlights the potential of vision-augmented molecular modeling.

8 Acknowledgment

This research has benefitted from the Microsoft Accelerating Foundation Models Research (AFMR) grant program.

References

[1] OpenAI, J. Achiam, and et. al., "Gpt-4 technical report," 2024.

[2] H. Liu, C. Li, Q. Wu, and Y. J. Lee, "Visual instruction tuning," 2023.

[3] D. Zhang, W. Liu, Q. Tan, J. Chen, H. Yan, Y. Yan, J. Li, W. Huang, X. Yue, W. Ouyang, D. Zhou, S. Zhang, M. Su, H.-S. Zhong, and Y. Li, "Chemllm: A chemical large language model," 2024.

[4] J. Li, D. Li, S. Savarese, and S. Hoi, "Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models," 2023.

[5] H. Liu, C. Li, Y. Li, and Y. J. Lee, "Improved baselines with visual instruction tuning," 2024.

[6] P. Gao, J. Han, R. Zhang, Z. Lin, S. Geng, A. Zhou, W. Zhang, P. Lu, C. He, X. Yue, H. Li, and Y. Qiao, "Llama-adapter v2: Parameter-efficient visual instruction model," 2023.

[7] W. Wang, Q. Lv, W. Yu, W. Hong, J. Qi, Y. Wang, J. Ji, Z. Yang, L. Zhao, X. Song, J. Xu, B. Xu, J. Li, Y. Dong, M. Ding, and J. Tang, "Cogvlm: Visual expert for pretrained language models," 2024.

[8] J. Bai, S. Bai, S. Yang, S. Wang, S. Tan, P. Wang, J. Lin, C. Zhou, and J. Zhou, "Qwen-vl: A versatile vision-language model for understanding, localization, text reading, and beyond," 2023.

[9] Q. Ye, H. Xu, G. Xu, J. Ye, M. Yan, Y. Zhou, J. Wang, A. Hu, P. Shi, Y. Shi, C. Li, Y. Xu, H. Chen, J. Tian, Q. Qian, J. Zhang, F. Huang, and J. Zhou, "mplug-owl: Modularization empowers large language models with multimodality," 2024.

[10] X. Chen, Z. Wu, X. Liu, Z. Pan, W. Liu, Z. Xie, X. Yu, and C. Ruan, "Janus-pro: Unified multimodal understanding and generation with data and model scaling," *arXiv preprint arXiv:2501.17811*, 2025.

[11] G. Landrum, "Rdkit documentation," *Release*, vol. 1, no. 1-79, p. 4, 2013.

[12] G. Subramanian, B. Ramsundar, V. Pande, and R. A. Denny, "Computational modeling of -secretase 1 (bace-1) inhibitors using ligand based approaches," *Journal of Chemical Information and Modeling*, vol. 56, no. 10, pp. 1936–1949, 2016. PMID: 27689393.

[13] O. T. Sakiyama H, Fukuda M, "Prediction of blood-brain barrier penetration (bbbp) based on molecular descriptors of the free-form and in-blood-form datasets. molecules.," 2021.

[14] Y. SS, "The nci's aids antiviral drug screening program," 1995.

[15] R. Huang, M. Xia, D.-T. Nguyen, T. Zhao, S. Sakamuru, J. Zhao, S. A. Shahane, A. Rossoshek, and A. Simeonov, "Tox21challenge to build predictive models of nuclear receptor and stress response pathways as mediated by exposure to environmental chemicals and drugs," *Frontiers in Environmental Science*, vol. 3, p. 85, 2016.

[16] J. S. Delaney, "Esol: estimating aqueous solubility directly from molecular structure," *Journal of chemical information and computer sciences*, vol. 44, no. 3, pp. 1000–1005, 2004.

[17] A. Karmaus, J. Fitzpatrick, D. Allen, G. Patlewicz, N. Kleinstreuer, and W. Casey, "Variability of ld50 values from rat oral acute toxicity studies: implications for alternative model development," *Society of Toxicology, San Antonio, TX*, vol. 3, pp. 11–15, 2018.

[18] A. Jain, S. P. Ong, G. Hautier, W. Chen, W. D. Richards, S. Dacek, S. Cholia, D. Gunter, D. Skinner, G. Ceder, *et al.*, "Commentary: The materials project: A materials genome approach to accelerating materials innovation," *APL materials*, vol. 1, no. 1, 2013.

[19] W. Hu, M. Fey, H. Ren, M. Nakata, Y. Dong, and J. Leskovec, "Ogb-lsc: A large-scale challenge for machine learning on graphs," *arXiv preprint arXiv:2103.09430*, 2021.

[20] C. Edwards, C. Zhai, and H. Ji, "Text2Mol: Cross-modal molecule retrieval with natural language queries," in *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing* (M.-F. Moens, X. Huang, L. Specia, and S. W.-t. Yih, eds.), (Online and Punta Cana, Dominican Republic), pp. 595–607, Association for Computational Linguistics, Nov. 2021.

[21] S. Lu, Z. Gao, D. He, L. Zhang, and G. Ke, "Highly accurate quantum chemical property prediction with uni-mol+," *arXiv preprint arXiv:2303.16982*, 2023.

[22] Z. Liu, S. Li, Y. Luo, H. Fei, Y. Cao, K. Kawaguchi, X. Wang, and T.-S. Chua, "Molca: Molecular graph-language modeling with cross-modal projector and uni-modal adapter," in *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 15623–15638, 2023.

[23] G. Helman, I. Shah, and G. Patlewicz, "Transitioning the generalised read-across approach (genra) to quantitative predictions: a case study using acute oral toxicity data," *Computational Toxicology*, vol. 12, p. 100097, 2019.

[24] C. Edwards, T. Lai, K. Ros, G. Honke, K. Cho, and H. Ji, "Translation between molecules and natural language," 2022.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: See Abstract and Section 1

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: See Appendix

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [NA]

Justification: NA

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: See Appendix and Code Repository

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: Code and Datasets are available at project page and github repository.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyper-parameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: Discussed in Appendix and documented in code repository

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: Discussed in the paper

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.

- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: See Appendix

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research adheres to the NeurIPS Code of Ethics

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: Discussed in the paper

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to

generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.

- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: NA

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [No]

    Justification: All datasets used are properly cited.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [Yes]

Justification: The code and datasets are released in the project page with proper guided and structured documentations.

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [NA]

Justification: NA

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [NA]

Justification: NA

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

Answer: [Yes]

Justification: Evaluation of LLMs for certain tasks

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.

Appendix

In this supplementary material, we begin with important discussions on limitations (Section A) and ethical considerations (Section B). We have also included additional dataset details (Section C), comprehensive results and analysis for classification tasks (Section D), regression tasks (Section E), and molecular description tasks (Section F). Furthermore, we present our approach to contrastive learning for vision encoders (Section G) and provide detailed prompt examples (Section H) supporting our discussion in the main paper. Figure 6 summarizes our multi modal approach for molecular property prediction.

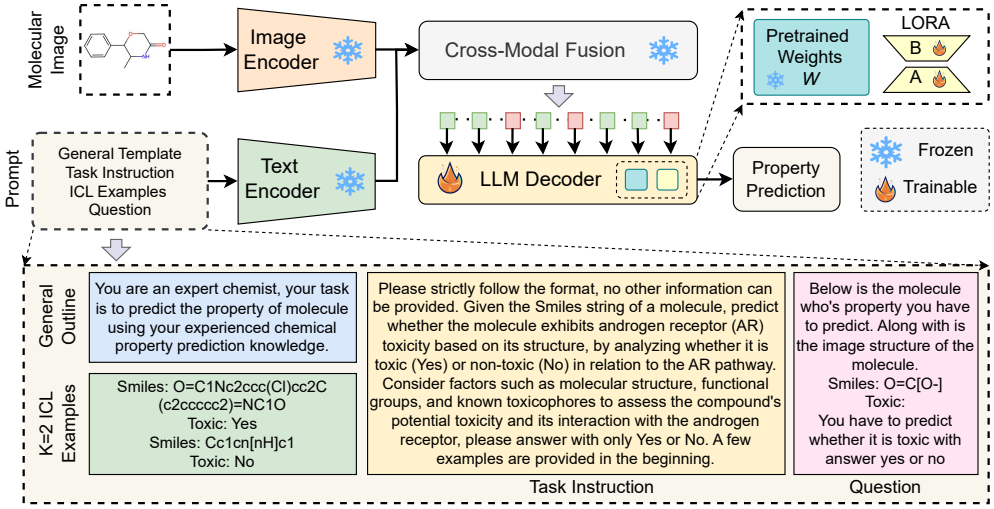

Figure 6: ***Overview of visual-textual approach for property prediction:*** The image representation along with textual description are used as input by the VLM where the image is encoded by a vision encoder and textual description is encoded by a text encoder. These multimodal features are used to generate the output with the help of a decoder. We show template prompt used for property prediction, including general outline, task instruction, in-context learning (ICL with k=2), and an image prompt.

A Limitations

Here we discuss some of the limitations of our work.

Adaptation of closed-source models: Our efficient adaptation of large visual language models for molecular property prediction is limited to open-source models. Considering the strong performance of proprietary models in case of few-shot learning, it will interesting to see how the capabilities of these closed-source models improve for this domain.

Advanced vision-language models: In our few-shot setup, we utilize an image representation of a molecule as additional input to the model. Since these models can take only one image as input, it was not possible to provide image representations for in-context examples as input. Future research could explore models capable of processing multiple images simultaneously.

B Ethical considerations

The integration of Vision-Language Models (VLMs) into molecular property prediction opens exciting new possibilities while also highlighting the importance of ethical considerations. By leveraging visual and textual representations, these models have the potential to accelerate discoveries in drug development and materials science in a more data-efficient manner. However, ensuring responsible AI development is crucial—focusing on model interpretability, fairness, and transparency can enhance trust and reliability in scientific applications. Additionally, proactive measures, such as open benchmarking and ethical guidelines, can help steer this technology toward positive societal impact while mitigating risks. By addressing these considerations thoughtfully, VLMs can become a transformative tool for chemistry and beyond.

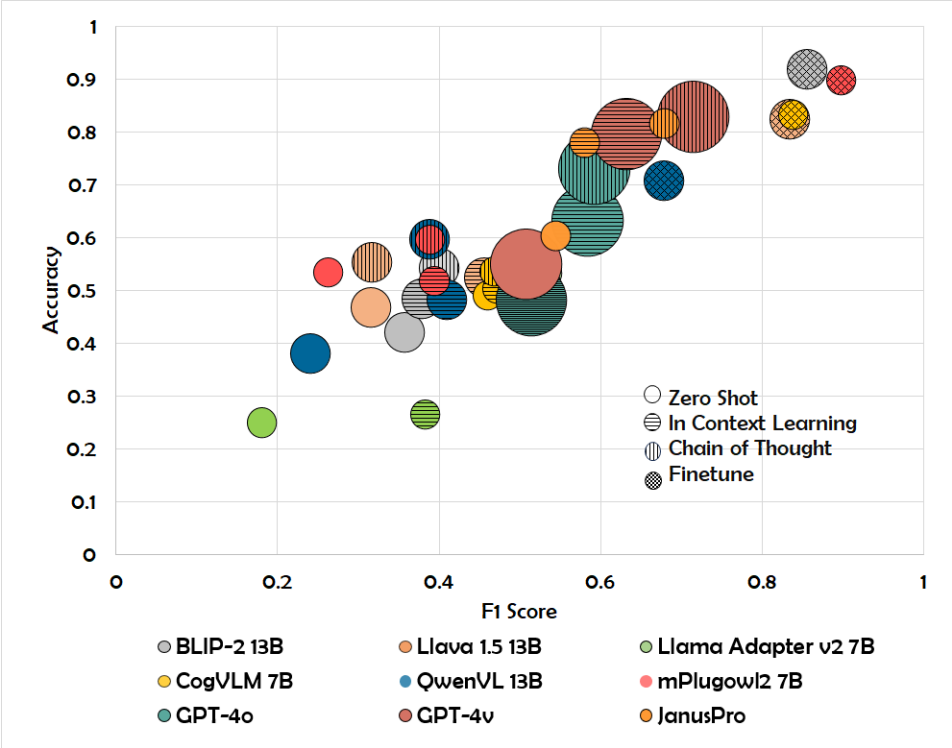

Figure 7: Performance comparison of Vision-Language Models (VLMs) across BACE, BBBP, HIV, Clintox, and Tox21 datasets, depicting Accuracy vs F1 Score. The bubble size represents the model's parameter scale

C Datasets: Additional Details

This study covers datasets with varied numbers of molecules, as low as 2k to as high as 3.7M. Figure 8 shows categorization of these datasets by tasks, Classification, Regression and Molecular Description. The default representation that is included with these datasets is SMILES and we generated the corresponding SELFIES representation and performed additional evaluations. All the data will be available on the provided link [1].

C.1 Model variants

VLMs are evolved from LLMs that are designed to understand visual information and generate language based on both, textual and visual inputs. They integrate vision encoders and natural language processing techniques to interpret and describe images enabling use cases such as image captioning, visual question answering and multimodal translation. We study nine different state-of-the-art visual language models in this study.

Janus-Pro 7B: Janus-Pro 7B is a unified multimodal model that employs a novel decoupled visual encoding approach. It processes visual inputs differently for understanding versus generation tasks: using a SigLIP encoder to extract semantic features for understanding, while employing a VQ tokenizer for generation. These distinct visual representations are then mapped through separate adaptors into a shared input space, where a 7B-parameter autoregressive transformer processes the combined multimodal sequences.
BLIP-2: BLIP-2 (Bootstrapping Language-Image Pre-training) is a multimodal model developed by Salesforce that combines visual and language modalities to improve performance on tasks involving both visual inputs and textual information generation.
Llava 1.5: It is a multimodal model that integrates text and image data, excelling in tasks like Visual Question Answering (VQA), image captioning, and cross-modal retrieval. The model uses Vicuna

[1] Code and datasets available at: `https://molvision.github.io/MolVision/`

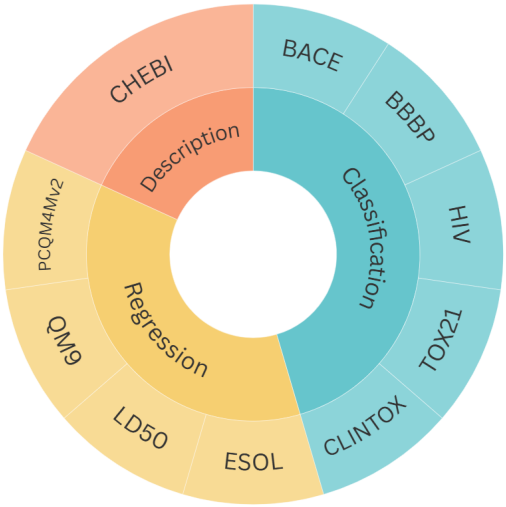

Figure 8: ***Distribution of Datasets by Task Type***: The chart illustrates the categorization of datasets based on their primary task, either classification (blue) or regression (yellow).

v1.5 as the base LLM.
Llama Adapter V2: The LLaMA-Adapter V2 is an adaption technique that is intended to improve the LLaMA model's ability to obey instructions while preserving parameter efficiency. It presents a number of important methods, such as early fusing of visual knowledge, joint training with discontinuous parameters, bias control of linear layers, and integration with expert models.
CogVLM: CogVLM is a vision-language model that integrates a Vision Transformer (ViT) encoder, MLP adapter, pretrained large language model, and a visual expert module. The ViT encoder uses the pretrained EVA2-CLIP-E model with the final layer removed for image feature compatibility.
QwenVL: Qwen-VL is a vision-language model for tasks like understanding, localization, and text reading. It consists of a visual encoder, a position-aware vision-language converter, and a large language model (Qwen-7B). The visual encoder, based on Openclip's ViT-bigG, processes images by dividing them into patches.
mPlugOwl 2: mPLUGOWL2 integrates a vision encoder, visual abstractor, and language decoder for vision-language tasks. The ViT-L/14 encoder processes images into visual tokens, which the LLaMA-2-7B decoder converts into text.
GPT-4V: GPT-4V (GPT-4 with Vision) advances multimodal AI by processing both visual and textual inputs. Though its architecture is proprietary, GPT-4V excels in understanding and describing images, solving visual problems, and performing detailed visual-language reasoning across various domains.
GPT-4o: GPT-4o, OpenAI's advanced language model, improves upon its predecessors with enhanced natural language processing, reasoning, and task completion. It offers better reliability, safety, and zero-shot generalization, though its architecture details remain largely undisclosed.

C.2 Task and Datasets

We utilize RDKit to generate molecular visualizations from SMILES structures. RDKit not only facilitates the conversion of SMILES strings into visual representations but also supports the transformation of SMILES into SELFIES strings. This functionality enables us to explore diverse molecular encoding techniques, thereby enhancing the robustness and adaptability of our predictive models. Since most existing datasets primarily feature SMILES strings, the ability to convert them to SELFIES representations extends the scope of our analysis. Each dataset contains a formatted prompt alongside the pathway to the visualized molecule image. Leveraging these datasets, we input the information into vision-language models for tasks such as visual question answering and instruction-based challenges. We use the following datasets in our benchmark.

BACE-V: The BACE-V dataset, adapted from the BACE (Binary Activity of Chemical Entities) dataset, provides 2D skeletal images of molecular structures along with key bioactivity data. Widely

used for binary classification in bioactivity prediction, particularly for BACE-1 inhibitors linked to Alzheimer's, the dataset includes both quantitative (IC50 values) and qualitative (binary) binding data. It features 154 BACE inhibitors for affinity prediction, 20 for pose prediction, and 34 for free energy prediction.

BBBP-V: The BBBP-V dataset, based on the Blood-Brain Barrier Penetration (BBBP) dataset, includes 2D skeletal images of molecular structures. It provides binary labels for BBB penetration (penetrant or non-penetrant) along with SMILES notations and key properties like molecular weight, lipophilicity (logP), and topological polar surface area (TPSA), all essential for predicting BBB permeability.

HIV-V: The HIV-V dataset, based on the HIV dataset, contains 2D skeletal images of molecular structures to support predictions of HIV replication inhibition. It includes binary labels for anti-HIV activity and key molecular parameters—molecular weight, logP, and TPSA—essential for assessing bioactivity and pharmacokinetics. Our evaluation focused on predicting HIV activity.

Clintox-V: The ClinTox-V dataset, derived from the ClinTox dataset, includes 2D skeletal images of molecular structures to support predictions of clinical toxicity and FDA approval status. Represented by SMILES notation, each of the 1,491 compounds is labeled for toxicity or FDA approval, enabling two classification tasks. Our evaluation focused on predicting FDA approval status.

Tox21-V: The Tox21-V dataset, based on the Tox21 dataset, includes 2D skeletal molecular images for predicting chemical toxicity, critical for environmental and pharmaceutical safety. It contains hundreds of compounds, each represented by SMILES notation, with twelve binary labels from toxicological tests. Our evaluation focused on the NR-AR binary label.

ESOL-V: The ESOL-V dataset, based on the ESOL (Estimating Solubility of Organic Compounds in Water) dataset, includes 2D skeletal molecular images and key data for predicting aqueous solubility of organic compounds, crucial for drug development and environmental studies.

LD50-V: It is based on the LD50 (Lethal Dose 50) dataset and includes 2D skeletal molecular images and data on acute toxicity. It focuses on the dose required to cause death in 50% of test subjects, a key metric for safety assessment in drug development and environmental health.

QM9-V: The QM9-V dataset, derived from the QM9 dataset, contains 2D skeletal representations of molecular structures in image format, alongside various quantum chemical properties. QM9 provides extensive data on 12 quantum mechanical properties, including the dipole moment, isotropic polarizability, electronic spatial extent, HOMO (Highest Occupied Molecular Orbital energy), LUMO (Lowest Unoccupied Molecular Orbital energy), and HOMO-LUMO gap, among others.

PCQM4Mv2-V: The PCQM4Mv2-V dataset, derived from the PCQM4Mv2 dataset, contains 2D skeletal molecular images paired with quantum property data. The dataset focuses on predicting the HOMO-LUMO gap, an essential quantum property that provides insights into a molecule's chemical stability and reactivity.

ChEBI-V: The ChEBI-V dataset, derived from the ChEBI database, contains 2D skeletal representations of molecular structures in image format, alongside comprehensive biological and chemical annotations. ChEBI-V provides structured information including molecular names, functional classifications, physicochemical properties, and biological roles such as enzyme inhibitors, receptor agonists, and therapeutic agents, making it valuable for molecular description and biological function prediction tasks.

D Classification: Further Analysis and Discussion

In this section, we discuss Zero-shot evaluation, effect of number of examples used in ICL,chain of thought prompting, effect of temperature in model performance and impact of visual data for classification task. Figure 7 show performance comparison of different models.

D.1 Finetuning Evaluation

Table 5 presents classification performance after LoRA finetuning across five molecular property prediction tasks. Vision-language models demonstrate superior performance compared to traditional machine learning approaches and domain-specific methods. BLIP-2 achieves the highest

average accuracy (0.92), excelling particularly on BBBP-V and Tox21-V tasks. mPlugOWL2 shows competitive performance with the best average F1 score (0.89) and strong results across multiple tasks. Traditional methods like XGBoost (0.85 average accuracy) and RF (0.83) provide reasonable baselines, while ChemLLM struggles significantly across all tasks (0.17 average). The results indicate that general-purpose vision-language models can effectively adapt to molecular classification tasks through parameter-efficient finetuning.

Table 5: ***Classification performance after finetuning:*** Accuracy (F1 score) comparison of models finetuned using LoRA. The best performing models are highlighted with bold text.

Models	**BACE-V** ↑	**BBBP-V**↑	**HIV-V** ↑	**ClinTox-V** ↑	**Tox21-V**↑	**Average** ↑
RF	0.79(0.76)	0.82(0.88)	0.87(0.52)	0.85(0.46)	0.83(0.26)	0.83(0.57)
XGBoost	0.81(0.77)	0.85(0.90)	0.87(0.55)	0.88(0.62)	0.84(0.33)	0.85(0.63)
ChemLLM (3)	0.18(0.12)	0.12(0.08)	0.19(0.09)	0.21(0.13)	0.18(0.09)	0.17(0.10)
Molca (22)	0.79(0.73)	0.74(0.72)	0.89(0.84)	0.93(0.84)	0.80(0.72)	0.83(0.77)
BLIP-2	**0.86(0.83)**	**0.93(0.96)**	**0.92**(0.76)	**0.89(0.93)**	**0.99**(0.80)	**0.92**(0.86)
Llava 1.5 13B	0.84(**0.83**)	0.86(0.88)	0.80(0.81)	0.70(0.72)	0.92(0.93)	0.82(0.83)
Llama Adapter v2 7B	0.52(0.48)	0.45(0.46)	0.43(0.42)	0.58(0.62)	0.68(0.69)	0.53(0.53)
CogVLM	0.72(0.71)	0.78(0.82)	0.85(0.83)	0.88(0.90)	0.93(0.93)	0.83(0.84)
Qwen VL	0.78(0.78)	0.70(0.72)	0.60(0.61)	0.71(0.64)	0.75(0.64)	0.71(0.68)
mPlugOWL2	**0.86**(0.82)	0.90(0.88)	0.90(**0.91**)	**0.89**(0.92)	0.94(**0.96**)	0.89(**0.89**)

D.2 Zero-Shot Evaluation

We have included more detailed results with Zero-shot performance (Table 6 and 7) where we only ask questions with general outline to the models without using any in-context examples.

Table 6: ***Zero-shot with SMILES:*** The table shows variation in the F1-score & accuracy of different models when subjected to zero-shot prompting. In this evaluation, only the basic instruction is provided to the models to predict whether a given molecule string is toxic or not, without any additional context or examples.

Model	**BACE-V**	**BBBP-V**	**HIV-V**	**Clintox-V**	**Tox21-V**
JanusPro 7B	0.45(0.37)	0.42(0.44)	0.52(0.32)	0.31(0.32)	0.44(0.34)
BLIP2	0.28 (0.29)	0.31 (0.29)	0.42 (0.33)	0.29 (0.28)	0.42 (0.31)
Llava 1.5	0.37 (0.54)	0.43 (0.46)	0.38 (0.35)	0.36 (0.39)	0.46 (0.39)
Llama	0.34 (0.39)	0.41 (0.28)	0.21 (0.33)	0.28 (0.31)	0.12 (0.13)
CogVLM	0.27 (0.34)	0.31 (0.32)	0.22 (0.25)	0.47 (0.49)	0.17 (0.12)
QwenVLM	0.32 (0.39)	0.29 (0.12)	0.22 (0.29)	0.22 (0.15)	0.45 (0.37)
mPlugowl2	0.39 (0.38)	0.32 (0.31)	0.41 (0.27)	0.27 (0.26)	0.67 (0.13)

Table 7: ***Zero-shot performance with SELFIES:*** The table illustrates the variation in the Accuracy (F1-score) of different models when subjected to zero-shot prompting. In this evaluation, only the basic instruction is provided to the models to predict whether a given molecule string is toxic or not, without any additional context or examples.

Model	**BACE-V**	**BBBP-V**	**HIV-V**	**ClinTox-V**	**Tox21-V**
JanusPro 7B	0.48(0.53)	0.54(0.581)	0.43 (0.36)	0.48 (0.37)	0.47 (0.31)
BLIP2	0.41 (0.34)	0.46 (0.48)	0.31 (0.29)	0.45 (0.47)	0.57 (0.21)
Llava 1.5	0.42 (0.48)	0.48 (0.50)	0.24 (0.33)	0.54 (0.64)	0.59 (0.15)
Llama Adapter v2 7B	0.42 (0.51)	0.39 (0.42)	0.21 (0.33)	0.41 (0.53)	0.14 (0.12)
CogVLM	0.42 (0.49)	0.49 (0.62)	0.28 (0.39)	0.44 (0.35)	0.16 (0.11)
QwenVL	0.42 (0.59)	0.41 (0.58)	0.21 (0.31)	0.16 (0.14)	0.16 (0.11)
mPlugOwl2	0.47 (0.25)	0.35 (0.23)	0.39 (0.28)	0.28 (0.22)	0.37 (0.17)

SMILES vs SELFIES: We examine and compare Zero-shot performance of models with SMILES and SELFIES representations. SELFIES generally yield better performance however on HIV dataset we see comparatively better performance with SMILES representation as shown in Tables 6 and 7. We also performed this analysis with ICL and has been discussed later.

D.3 Effect of ICL Examples

We conducted a comprehensive analysis of effects of number of examples (k = 0, 2, 4) in in-context learning (ICL) across vision-language models for molecular property prediction. More context does not always yield better results (Table 8, 9). The effectiveness of ICL varies significantly across datasets, as evidenced by CogVLM's substantial improvement on ClinTox-V when increasing from k = 0 to k = 4 (0.54 to 0.76 accuracy). We also observed similar behavious with CogVLM using SELFIES in Table 11. Different models demonstrate varying sensitivity to ICL, BLIP-2 however show consistent improvement with increased context, achieving its best performance with k = 4 across most datasets (Table 10).

Model QwenVL shows peak performance with k = 2 on several datasets (Table 14). Comparing SMILES representations (Table 8, 9) versus SELFIES representations (Table 12, 13,14 15, 16), SELFIES maintains more stable performance across different k values, particularly for complex models like mPlugOwl2 and CogVLM. These findings indicate that ICL's effectiveness depends heavily on model architecture, molecular representation, and dataset characteristics.

We also included results with increased in-context examples with gpt-4o. With the exception of BBBP-V accuracy improved across all datasets with an increase in the number of in-context examples (k) to six or eight. Notably, on the BACE-V, and Clintox-V datasets, we observed approximately a 40% increase in accuracy. With the exception of BACE-V, the F1-score was also highest at k=2 or 4 across all datasets (Table 17).

Table 8: ***Role of in-context examples:*** ICL with k = 0 showing Accuracy (F1-score) of various models on different datasets with SMILES representations.

Model	BACE-V	BBBP-V	HIV-V	ClinTox-V	Tox21-V
JanusPro 7B	0.65(0.68)	0.52(0.51)	0.92(0.68)	0.41(0.31)	0.52(0.44)
BLIP2	0.36 (0.52)	0.33 (0.27)	0.41 (0.36)	0.37 (0.28)	0.63 (0.29)
Llava 1.5	0.55 (0.18)	0.47 (0.43)	0.35 (0.32)	0.33 (0.33)	0.64 (0.11)
Llama	0.39 (0.59)	0.36 (0.42)	0.21 (0.33)	0.22 (0.19)	0.15 (0.19)
CogVLM	0.39 (0.56)	0.48 (0.48)	0.39 (0.26)	0.54 (0.54)	0.65 (0.11)
QwenVLM	0.41 (0.52)	0.31 (0.11)	0.28 (0.21)	0.38 (0.12)	0.52 (0.43)
mPlugowl2	0.48 (0.16)	0.41 (0.43)	0.65 (0.18)	0.28 (0.28)	0.85 (0.11)

Table 9: ***Effect of in-context examples:*** ICL with k = 4 showing Accuracy (F1-score) of various models on different datasets with SMILES Representation.

Model	BACE-V	BBBP-V	HIV-V	ClinTox-V	Tox21-V
JanusPro 7B	0.73(0.62)	0.63(0.601))	0.95(0.69)	0.97(0.64)	0.71(0.62)
BLIP2	0.29 (0.42)	0.19 (0.11)	0.52 (0.32)	0.34 (0.36)	0.55 (0.39)
Llava 1.5	0.48 (0.38)	0.57 (0.66)	0.32 (0.33)	0.25 (0.22)	0.64 (0.25)
Llama	0.39 (0.56)	0.37 (0.22)	0.21 (0.33)	0.32 (0.19)	0.28 (0.12)
CogVLM	0.39 (0.54)	0.64 (0.34)	0.39 (0.26)	0.68 (0.48)	0.19 (0.11)
QwenVLM	0.42 (0.52)	0.31 (0.11)	0.42 (0.54)	0.81 (0.12)	0.72 (0.17)
mPlugowl2	0.58 (0.42)	0.43 (0.38)	0.71 (0.25)	0.38 (0.42)	0.83 (0.13)

Table 10: ***Effect of in-context examples:*** Accuracy (F1-score) of BLIP-2 Model using SELFIES representations with variation in number of in-context examples used in the prompt (k = 0, 2, 4).

Variation	BACE-V	BBBP-V	HIV-V	ClinTox-V	Tox21-V
k = 0	0.43(0.26)	0.38(0.27)	0.54(0.32)	0.33(0.42)	0.52(0.21)
k = 2	0.36**(0.52)**	0.37(0.29)	0.60(0.29)	0.34(0.36)	0.75(0.42)
k = 4	**0.61**(0.27)	**0.39(0.31)**	**0.81(0.39)**	**0.36(0.44)**	**0.79(0.48)**

D.4 Chain of Thought Prompting

Table 18 demonstrates the effectiveness of Chain of Thought (CoT) prompting on molecular property classification tasks across five benchmark datasets. GPT-4v emerges as the top performer with an

Table 11: ***Impact of number of in-context examples:*** The table illustrates the variation in the performance of the CogVLM model with ICL in terms of Accuracy (F1-score), which utilizes the Vicuna 7B as its backbone, when tested on the SELFIE representation of various datasets. The performance is evaluated with different numbers of in-context examples (k = 0, 2, 4) provided in the prompt. The following results are produced with temperature set to 0.

Variation	BACE-V	BBBP-V	HIV-V	ClinTox-V	Tox21-V
k = 0	0.34 (0.51)	0.36 (0.32)	0.24 (0.33)	0.54 (0.65)	0.26 (0.15)
k = 2	0.62 (0.44)	0.51 (0.58)	0.25 (0.29)	0.32 (0.37)	0.18 (0.14)
k = 4	0.44 (0.53)	0.38 (0.45)	0.32 (0.31)	0.76 (0.86)	0.47 (0.13)

Table 12: ***Impact of number of in-context examples:*** The table illustrates the variation in the performance of the Llava 1.5 model with ICL in terms of Accuracy (F1-score) using Llava 1.5 13 billion parameters, when tested on the SELFIE representation of various datasets. The performance is evaluated with different numbers of in-context examples (k = 0, 2, 4) provided in the prompt. (Temperature=0).

Variation	BACE-V	BBBP-V	HIV-V	ClinTox-V	Tox21-V
k = 0	0.62 (0.24)	0.35 (0.17)	0.51 (0.13)	0.20 (0.14)	0.76 (0.17)
k = 2	0.61 (0.33)	0.73 (0.46)	0.32 (0.35)	0.35 (0.36)	0.67 (0.47)
k = 4	0.49 (0.38)	0.56 (0.29)	0.42 (0.33)	0.25 (0.19)	0.89 (0.11)

Table 13: ***Impact of number of in-context examples:*** The table illustrates the variation in the performance of the mPlugOwl2 model with ICL in terms of Accuracy (F1-score). This model utilizes Llama2 7B as its backbone and is tested on the SELFIE representation of various datasets. The performance is evaluated with different numbers of in-context examples (k = 0, 2, 4) provided in the prompt. (Temperature = 0).

Variation	BACE-V	BBBP-V	HIV-V	ClinTox-V	Tox21-V
k = 0	0.53 (0.22)	0.48 (0.52)	0.57 (0.19)	0.22 (0.15)	0.62 (0.17)
k = 2	0.64 (0.65)	0.46 (0.46)	0.76 (0.74)	0.39 (0.43)	0.74 (0.21)
k = 4	0.61 (0.31)	0.35 (0.36)	0.73 (0.41)	0.46 (0.57)	0.76 (0.14)

Table 14: ***Impact of number of in-context examples:*** The table illustrates the variation in the performance of the QwenVL model with ICL in terms of Accuracy (F1-score) using QwenVL 7 B parameters, when tested on the SELFIE representation of various datasets. The performance is evaluated with different numbers of in-context examples (k = 0, 2, 4) provided in the prompt.(Temperature=0).

Variation	BACE-V	BBBP-V	HIV-V	ClinTox-V	Tox21-V
k = 0	0.41 (0.52)	0.31 (0.10)	0.80 (0.11)	0.18 (0.12)	0.52 (0.14)
k = 2	0.45 (0.38)	0.63 (0.49)	0.79 (0.49)	0.50 (0.48)	0.78 (0.76)
k = 4	0.42 (0.51)	0.29 (0.09)	0.81 (0.10)	0.42 (0.54)	0.72 (0.17)

Table 15: ***Impact of number of in-context examples*** The table illustrates the variation in the Accuracy (F1-score) of the ICL model using Llama Adapter V2, which utilizes Llama2 7B as its backbone, when tested on the SELFIE representation of various datasets. The performance is evaluated with different numbers of in-context examples (k = 0, 2, 4) provided in the prompt.(Temperature=0).

Variation	BACE-V	BBBP-V	HIV-V	ClinTox-V	Tox21-V
k = 0	0.37 (0.34)	0.49 (0.21)	0.21 (0.33)	0.30 (0.39)	0.14 (0.17)
k = 2	0.38 (0.44)	0.48 (0.23)	0.21 (0.37)	0.31 (0.35)	0.15 (0.17)
k = 4	0.36 (0.35)	0.51 (0.27)	0.31 (0.42)	0.50 (0.31)	0.15 (0.18)

Table 16: ***Impact of number of in-context examples:*** The table illustrates the variation in the Accuracy (F1-score) of the ICL model using the BLIP-2 model when tested on the SELFIE representation of various datasets. The performance is evaluated with different numbers of in-context examples (k = 0, 2, 4) provided in the prompt.(Temperature=0).

Variation	BACE-V	BBBP-V	HIV-V	Clintox-V	Tox21-V
k = 0	0.56 (0.10)	0.31 (0.12)	0.49 (0.24)	0.16 (0.09)	0.47 (0.32)
k = 2	0.61 (0.15)	0.35 (0.16)	0.56 (0.24)	0.18 (0.12)	0.41 (0.22)
k = 4	0.66 (0.26)	0.32 (0.11)	0.51 (0.29)	0.21 (0.10)	0.45 (0.19)

Table 17: **Effect of in-context examples:** Accuracy (F1-score) for different ICL examples on GPT-4o model.

ICL Variation	BACE-V	BBBP-V	HIV-V	Clintox-V	Tox21-V	Average
k=0	0.39 (**0.55**)	0.56 (0.64)	0.72 (0.53)	0.25 (0.33)	0.49 (0.46)	0.48/0.50
k=2	0.56 (0.53)	**0.77** (**0.81**)	0.82 (0.56)	0.59 (0.44)	0.42 (**0.58**)	0.63/0.58
k=4	0.64 (0.52)	0.63 (0.66)	0.82 (0.78)	0.71 (**0.69**)	0.52 (0.44)	0.66/**0.61**
k=6	0.61 (0.48)	0.56 (0.62)	**0.86** (**0.79**)	**0.76** (0.63)	**0.61** (0.43)	**0.68**/0.59
k=8	**0.72** (0.51)	0.56 (0.60)	0.72 (0.64)	0.67 (**0.69**)	0.55 (0.34)	0.64/0.55
k=10	0.55 (0.35)	0.55 (0.59)	0.69 (0.23)	0.49 (0.53)	**0.61** (0.27)	0.57/0.39

average accuracy of 72.32%, closely followed by GPT-4o at 71.14%, indicating that both commercial models excel when employing structured reasoning approaches. Janus achieves the third-highest performance with an average accuracy of 71.60%, demonstrating competitive capabilities among open-source models and particularly excelling on HIV-V (93.3%) and ClinTox-V (97.2%) datasets.

The results reveal significant performance variations across datasets, with HIV-V generally showing the highest accuracy scores across models, while BBBP-V and Tox21-V present greater challenges. Notably, QwenVLM shows strong performance on HIV-V (82.9%) and Tox21-V (73.9%), while mPlugOWL2 demonstrates exceptional performance on specific datasets like Tox21-V (76.0%) and HIV-V (75.2%). The remaining models exhibit moderate performance, with CogVLM achieving balanced results across datasets and BLIP-2 showing consistent but lower performance.

Table 18: ***Classification Performance in Chain of Thought Prompting:*** Comparisons of models evaluated on classification tasks across various datasets using Chain-of-Thought (CoT) prompting showing Accuracy (F1-score) with SMILES representations.

Model	BACE-V	BBBP-V	HIV-V	ClinTox-V	Tox21-V
GPT-4o	0.783(0.612)	0.696(0.481)	0.893(0.829)	0.683(0.582)	0.601(0.455)
GPT-4v	0.800(0.749)	0.716(0.620)	0.928(0.842)	0.972(0.729)	0.728(0.632)
BLIP-2	0.49 (0.52)	0.49 (0.41)	0.62 (0.32)	0.54 (0.36)	0.57 (0.39)
CogVLM	0.510(0.559)	0.673(0.396)	0.422(0.384)	0.650(0.701)	0.430(0.303)
mPlugOWL2	0.716(0.414)	0.555(0.362)	0.752 (0.413)	0.461(0.574)	0.76 (0.148)
Llava	0.523 (0.462)	0.582(0.493)	0.42 (0.33)	0.352(0.19)	0.893(0.11)
Llama-Adapter	0.430(0.339)	0.554(0.674)	0.429(0.312)	0.684(0.712)	0.437(0.382)
QwenVLM	0.528(0.429)	0.394(0.291)	0.829(0.329)	0.492(0.421)	0.739(0.471)
Janus	0.797(0.669)	0.696(0.661)	0.933(0.781)	0.972(0.729)	0.681(0.557)

D.5 Effect of Temperature

The effect of temperature variation was shown in the main paper on one model (BLIP2 model) and here we include more results examining the effect of temperature variation under different settings (Table 21, 19 Table 24, and Table 23, 20, 22). We analyzed the impact of sampling temperature (ranging from 0.0 to 0.8) on model performance across different molecular representations and architectures. For SELFIES representation, the Llama Adapter v2 model shows optimal performance at moderate temperatures (0.2-0.4) for the BBBP-V dataset, achieving accuracy of 0.66 at temperature 0.2 (Table 20). SMILES representation exhibits different temperature sensitivity, with generally

improved performance at higher temperatures across datasets (Table 21). BLIP2 demonstrates consistent improvement with increasing temperature, achieving peak average performance of 0.51 at temperature 0.6 (Table 19). Llava 1.5 13B shows optimal performance at lower temperatures, particularly for the Tox21-V dataset with 0.93 accuracy at temperature 0.2 (Table 23). CogVLM exhibits more stable performance across temperature variations, with slight degradation at higher temperatures (Table 24). The mPlugOWL2 model achieves its best performance at temperature 0.2 across multiple datasets, notably reaching 0.88 accuracy on Tox21-V (Table 19). These findings suggest that moderate temperatures (0.2-0.4) generally provide optimal performance across models and representations, with specific optimal values being model and dataset dependent.

Table 19: **Effect of temperature:** Accuracy (F1-score) for different temperature settings using ICL (Samples k=2) on mPlugOWL2 model with SMILES.

Temp Variation	BACE-V	BBBP-V	HIV-V	Clintox-V	Tox21-V
0.0	0.59(**0.32**)	0.35(0.38)	0.62(**0.30**)	0.34(0.42)	0.69(**0.57**)
0.2	**0.70**(0.28)	0.38(0.24)	**0.74**(0.18)	**0.78**(0.18)	**0.88**(0.10)
0.4	0.65(0.15)	**0.46(0.46)**	0.64(0.21)	0.30(0.36)	0.83(0.16)
0.6	0.60(0.23)	0.43(0.44)	0.62(0.26)	0.40(0.51)	0.73(0.16)
0.8	0.58(0.19)	0.41(0.43)	0.59(0.28)	0.43(**0.52**)	0.72(0.20)

Table 20: ***Effect of temperature on SELFIE:*** This table shows the performance of the Llama Adapter v2 model with ICL (k=2 examples) on various datasets represented in SELFIE notation.

Temp	BACE-V	BBBP-V	HIV-V	Clintox-V	Tox21-V
0.0	0.38 (0.44)	0.48 (0.23)	0.21 (0.37)	0.31 (0.35)	0.15 (0.17)
0.2	0.34 (0.51)	0.66 (0.79)	0.19 (0.31)	0.29 (0.88)	0.13 (0.18)
0.4	0.35 (0.49)	0.62 (0.75)	0.22 (0.32)	0.24 (0.85)	0.28 (0.11)
0.6	0.36 (0.43)	0.53 (0.68)	0.26 (0.33)	0.22 (0.83)	0.31 (0.18)
0.8	0.33 (0.44)	0.58 (0.69)	0.24 (0.39)	0.22 (0.81)	0.31 (0.15)

Table 21: ***Effect of temperature on SMILES:*** This table shows the performance of the Llama Adapter v2 model with ICL (k=2 examples) on various datasets represented in SMILES notation.

Temp	BACE-V	BBBP-V	HIV-V	Clintox-V	Tox21-V
0.0	0.28 (0.29)	0.18 (0.11)	0.19 (0.17)	0.29 (0.12)	0.31 (0.21)
0.2	0.38 (0.55)	0.22 (0.83)	0.21 (0.33)	0.28 (0.19)	0.32 (0.15)
0.4	0.35 (0.49)	0.35 (0.76)	0.42 (0.35)	0.22 (0.24)	0.39 (0.24)
0.6	0.39 (0.56)	0.23 (0.74)	0.31 (0.25)	0.31 (0.11)	0.41 (0.29)
0.8	0.43 (0.54)	0.29 (0.62)	0.37 (0.36)	0.26 (0.19)	0.41 (0.11)

Table 22: **Effect of Temperature on Model Performance:** Accuracy (F1-score) at different temperature settings using the BLIP2 model on various datasets. Higher temperatures generally show increased variability in F1-scores, with peak performance occurring at different temperature levels across datasets.

Temp Variation	BACE-V	BBBP-V	HIV-V	Clintox-V	Tox21-V	Average
0.0	0.33(0.42)	0.29(0.27)	0.56(0.24)	0.28(0.23)	0.65(0.32)	0.42(0.30)
0.2	0.35(0.49)	0.30(0.28)	0.56(0.30)	0.29(0.16)	0.69(0.34)	0.44(0.31)
0.4	0.34(0.42)	0.32(0.28)	0.59(0.27)	0.32(0.34)	0.72(0.34)	0.46(0.33)
0.6	0.41(0.54)	0.39(0.31)	0.64(0.35)	0.38(0.38)	0.72(0.36)	0.51(0.39)
0.8	0.38(0.48)	0.36(0.24)	0.62(0.32)	0.38(0.26)	0.78(0.44)	0.50(0.35)

Table 23: ***Effect of temperature:*** Performance analysis using the Llava 1.5 13B with ICL (k=2), focusing on diverse datasets represented in SELFIE format.

Temp	BACE-V	BBBP-V	HIV-V	Clintox-V	Tox21-V
0.0	0.61 (0.33)	0.73 (0.46)	0.32 (0.35)	0.35 (0.36)	0.67 (0.47)
0.2	0.59 (0.34)	0.63 (0.18)	0.36 (0.14)	0.19 (0.14)	0.93 (0.11)
0.4	0.57 (0.41)	0.34 (0.25)	0.34 (0.25)	0.31 (0.11)	0.86 (0.12)
0.6	0.48 (0.29)	0.42 (0.43)	0.36 (0.43)	0.34 (0.34)	0.77 (0.18)
0.8	0.52 (0.41)	0.37 (0.35)	0.30 (0.35)	0.36 (0.13)	0.75 (0.07)

Table 24: ***Effect of temperature:*** Analysis of temperature variation in various SELFIE based datasets with ICL (k=2) using CogVLM model.

Temp	BACE-V	BBBP-V	HIV-V	Clintox-V	Tox21-V
0.0	0.62 (0.44)	0.51 (0.58)	0.25 (0.29)	0.32 (0.37)	0.18 (0.14)
0.2	0.58 (0.41)	0.48 (0.54)	0.28 (0.52)	0.32 (0.31)	0.14 (0.35)
0.4	0.58 (0.41)	0.42 (0.52)	0.26 (0.48)	0.33 (0.33)	0.17 (0.34)
0.6	0.53 (0.44)	0.45 (0.48)	0.25 (0.42)	0.27 (0.31)	0.14 (0.32)
0.8	0.56 (0.39)	0.37 (0.39)	0.28 (0.49)	0.25 (0.30)	0.18 (0.32)

D.6 Impact of visual data

Table 25 underscores the stark contrast in performance between Llama 2, a large language model, and its VLM counterpart, Llama Adapter v2 after ICL. Llama Adapter v2 also show substantial improvement post-finetuning.

Table 25: ***Impact of visual data:*** First row shows Accuracy (F1-score) for ICL with language model Llama2, second row shows visual-language variant with improvement in performance, and third row demonstrates significant improvement in performance after finetuning.

Models	BACE-V	BBBP-V	HIV-V	ClinTox-V	Tox21-V
Llama 2 13B (ICL)	<0.01(<0.01)	0.05(0.04)	0.05(0.07)	0.05(0.08)	<0.01(<0.01)
Llama Adapter v2 7B (ICL)	0.28(0.29)	0.18(0.11)	0.19(0.17)	0.29(0.12)	0.31(0.21)
Llama Adapter v2 7B (LoRA)	0.52(0.48)	0.45(0.46)	0.43(0.42)	0.58(0.62)	0.68(0.69)

E Regression: Further Analysis and Discussion

The main paper presents the results of regression tasks on ESOL, LD50, QM9, and PCQM4Mv2 for in-context learning (ICL) with $k = 2$ and finetuning. Here we provide additional results for **zero-shot** learning, detailed in Table 27, and **few-shot** learning with $k = 4$, shown in Table 28. Figure 9 summarizes performance comparison across different models for regression tasks.

Furthermore, comprehensive evaluations on the QM9 dataset are included for 12 quantum mechanical targets under the **"all-together"** setting, where all targets are prompted simultaneously. These evaluations cover zero-shot learning ($k = 0$), shown in Table 29; few-shot learning with $k = 2$, presented in Table 30; and few-shot learning with $k = 4$, detailed in Table 31.

E.1 Effect of Finetuning

The performance improvements achieved through LoRA-based finetuning are substantial across all regression tasks, as demonstrated in Table 26. BLIP-2 achieves the best overall performance with an average error of 1.925 across all datasets, excelling particularly on QM9-V (4.923 MAE) and PCQM4Mv2-V (0.235 MAE). CogVLM follows closely with an average error of 1.928, showing exceptional performance on ESOL-V with an RMSE of 1.102. Notably, Qwen VL demonstrates remarkable accuracy on LD50-V with a minimal MAE of 0.022. These finetuning results significantly outperform the in-context learning (ICL k=2) approaches presented in Table 3 of the main paper,

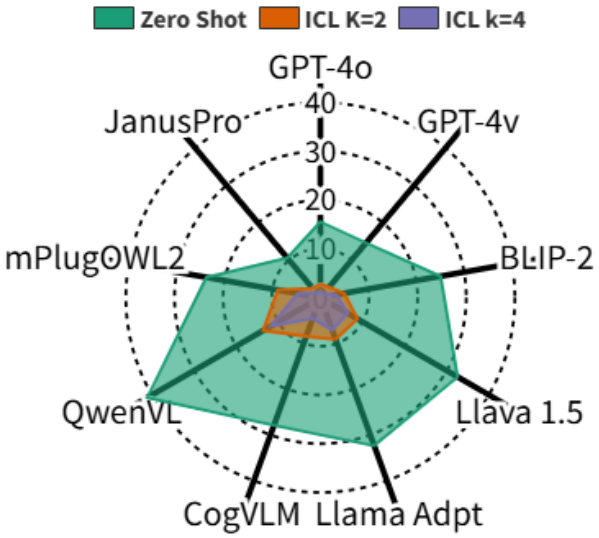

Figure 9: Radar plot comparing regression performance across various models (GPT-4v, GPT-4o, JansuPro, QwenVL, mPlugOWL2, BLIP-2, Llava 1.5 13B, CogVLM, and Llama Adapter v2 7B) averaged across ESOL, LD50, QM9, and PCQM4Mv2 datasets. The chart highlights Zero Shot (green) and Few Shot (k=2 in orange, k=4 in purple) capabilities.

where even the best-performing ICL model, Janus-Pro 7B, achieved only 2.52 average error. For example, BLIP-2's finetuned performance (1.925 average error) represents a 61.5% improvement over its ICL performance (5.01 average error). Similarly, CogVLM improved from 8.50 to 1.928, and Qwen VL from 13.64 to 2.340. These dramatic improvements underscore the limitations of few-shot learning for molecular property prediction tasks and highlight the critical importance of task-specific parameter adaptation through finetuning.

Table 26: ***Performance comparison after finetuning:*** Regression tasks. Error comparison of models finetuned using LoRA across different datasets. The best performing models are highlighted with bold text. Second best model performance are underlined.

Model	ESOL-V (RMSE)	LD50-V (MAE)	QM9-V (MAE)	PCQM4Mv2-V (MAE)	Average
BLIP-2	1.764	0.779	**4.923**	**0.235**	**1.925**
Llava 1.5 13B	2.229	0.193	5.193	<u>0.602</u>	2.554
Llama Adapter v2 7B	4.032	0.624	7.921	3.002	3.895
CogVLM	**1.102**	0.592	5.221	0.795	<u>1.928</u>
Qwen VL	2.192	**0.022**	<u>5.021</u>	2.125	2.340
mPlugOWL2	<u>1.291</u>	<u>0.082</u>	8.029	1.621	2.756

E.2 Effect of ICL Examples

Tables 27 and 28 shows significant impact of in-context examples (k=4) versus zero-shot learning (k=0) across all models. GPT-4v maintains superior performance in both scenarios, ranking first with average metrics of 14.6485 and 1.04825 respectively, as shown in Table 27 and Table 28. The introduction of examples leads to substantial error reduction, exemplified by GPT-4v's ESOL RMSE improving from 1.489 to 0.812, and its QM9 MAE decreasing from 45.296 to 2.503. Notably, the performance gap between models narrows with in-context examples, as evidenced by the reduction in average performance difference between best and worst models from 26.4815 (Table 27) to 10.69325 (Table 28), particularly in complex tasks like QM9 and PCQM4Mv2.

Table 27: ***Performance comparison for zero-shot learning (k=0):*** MAE, RMSE of Multimodal LLMs for molecular property prediction based regression tasks.

Model	ESOL (RMSE)	LD50 (MAE)	QM9 (MAE)	PCQM4Mv2 (MAE)	Average
GPT-4o	**1.232**	12.31	47.126	**1.41**	15.5195
GPT-4v	1.489	9.01	**45.296**	2.799	**14.6485**
JanusPro 7B	1.997	**8.124**	48.301	1.893	15.07875
BLIP-2	2.011	15.631	78.731	3.973	25.0865
Llava 1.5	18.198	14.952	93.182	3.967	32.57475
Llama Adapter	3.314	5.741	106.382	14.005	32.3605
CogVLM	2.093	23.769	80.766	4.939	27.89175
Qwen	4.119	24.388	116.353	19.66	41.13
mPlugOWL2	2.12	10.888	77.297	4.899	23.801

Table 28: ***Performance comparison for few-shot learning (k=4):*** MAE, RMSE of Multimodal LLMs for molecular property prediction based regression tasks.

Model	ESOL (RMSE)	LD50 (MAE)	QM9 (MAE)	PCQM4Mv2 (MAE)	Average
GPT-4o	0.867	**0.614**	3.147	0.222	1.2125
GPT-4v	0.812	0.686	**2.503**	**0.192**	**1.048**
JanusPro 7B	**0.562**	0.632	3.071	0.348	1.421
BLIP-2	1.289	0.696	10.339	0.882	3.3015
Llava 1.5	4.361	0.709	20.157	0.883	6.5275
Llama Adapter	2.309	2.431	19.840	3.809	7.09725
CogVLM	1.225	0.815	15.662	0.805	4.62675
Qwen	3.332	0.839	33.534	9.261	11.7415
mPlugOWL2	1.416	0.741	14.998	1.692	4.71175

E.3 Analysis of QM9 Multi-Target Prediction

Tables 29, 30, and 31 present the performance comparison for simultaneous prediction of all 12 QM9 molecular properties. In the zero-shot setting (Table 29), GPT-4o and GPT-4v demonstrate superior performance with average MAEs of 47.1264 and 45.2964 respectively. The addition of in-context examples (k=2, k=4) significantly improves prediction accuracy across all models, with GPT-4v achieving the best average MAE of 2.5028 at k=4. Notably, both GPT-4o and GPT-4v show consistent performance across different molecular properties, maintaining their superiority even in this challenging multi-target prediction scenario.

Table 29: ***Performance comparison for zero-shot learning (k=0):*** MAE of Multimodal LLMs for molecular property prediction based regression tasks on QM9 Dataset.

Model	QM9(Alpha)	QM9(Gap)	QM9(Homo)	QM9(Lumo)	QM9(Mu)	QM9(CV)	QM9(G298)	QM9(H298)	QM9(r2)	QM9(u298)	QM9(u0)	QM9(zpve)	QM9(Avg)
GPT-4o	**11.0855**	**1.8364**	**1.0287**	3.0245	**2.6813**	6.7424	**85.0988**	**85.0906**	**195.5142**	**85.0902**	**85.0929**	3.2317	**47.1264**
GPT-4v	14.925	2.4481	1.9924	**2.129**	3.129	6.1294	88.924	120.449	145.9812	66.2498	88.9192	**2.2804**	45.2964
JanusPro 7B	12.921	3.8238	1.8299	2.921	3.109	4.291	97.921	91.842	170.829	73.289	111.924	4.921	48.301
BLIP-2	47.902	3.920	2.220	2.844	11.294	59.201	122.842	129.912	255.901	102.120	194.201	12.422	78.7316
Llava 1.5	108.449	32.923	21.382	17.994	17.544	36.686	173.561	126.244	209.361	186.335	157.421	30.285	93.1821
Llama-Adapter	175.989	13.822	11.939	15.679	12.525	79.738	207.088	130.005	232.172	165.611	201.434	30.584	106.3822
CogVLM	93.191	2.119	1.4757	2.997	9.784	45.534	137.238	184.106	232.069	132.701	120.716	7.2616	80.7660
Qwen	184.191	8.201	7.772	9.009	46.111	137.302	240.019	129.090	220.322	172.828	197.466	43.923	116.3528
mPlugOWL2	72.706	3.676	4.280	2.457	5.673	31.363	157.058	157.078	214.541	100.495	121.920	56.315	77.2968

F Molecular Description: Further Analysis

We conducted a comprehensive evaluation of molecular description capabilities across multiple models and settings. This analysis examines zero-shot performance, the effects of in-context learning (ICL) with varying numbers of examples, and the impact of Chain of Thought (CoT) prompting on

Table 30: ***Performance comparison for few-shot learning (k=2):*** MAE of Multimodal LLMs for molecular property prediction based regression tasks on QM9 Dataset.

Model	QM9(Alpha)	QM9(Gap)	QM9(Homo)	QM9(Lumo)	QM9(Mu)	QM9(CV)	QM9(G298)	QM9(H298)	QM9(r2)	QM9(u298)	QM9(u0)	QM9(zpve)	QM9(Avg)
GPT-4o	**2.32**	**0.2126**	**0.3113**	**0.2212**	0.9787	**0.857**	**17.9428**	**17.9434**	**22.948**	**17.9435**	**17.9432**	0.9126	**8.3779**
GPT-4v	3.1	0.484	0.396	0.293	**0.968**	0.998	18.021	18.022	24.405	18.022	18.021	**0.725**	8.6217
JanusPro 7B	2.892	0.429	0.3329	0.3392	0.792	0.729	18.291	18.728	21.882	19.821	17.211	0.884	8.5276
BLIP-2	3.401	0.713	0.891	0.629	2.14	2.129	22.092	21.921	85.239	29.912	22.12	0.982	16.014
Llava 1.5	17.472	2.191	1.991	1.629	7.737	3.516	26.653	68.502	100.307	56.087	36.873	1.024	26.9985
Llama-adapter	17.316	1.621	2.091	1.032	4.554	5.011	39.585	45.626	130.37	58.654	29.255	2.02	28.0946
CogVLM	7.6465	0.91	1.012	0.876	3.613	2.799	21.292	27.702	174.554	45.178	23.74	0.822	25.8454
QwenVLM	13.563	2.03	1.293	1.075	10.587	3.563	130.992	75.385	134.581	55.007	36.941	2.036	38.9211
mPlugOWL2	10.336	1.832	1.432	1.71	2.685	3.233	23.091	37.263	189.806	52.405	27.379	0.829	29.333

Table 31: ***Performance comparison for few-shot learning (k=4):*** MAE of Multimodal LLMs for molecular property prediction based regression tasks on QM9 Dataset.

Model	QM9(Alpha)	QM9(Gap)	QM9(Homo)	QM9(Lumo)	QM9(Mu)	QM9(CV)	QM9(G298)	QM9(H298)	QM9(r2)	QM9(u298)	QM9(u0)	QM9(zpve)	QM9(Avg)
GPT-4o	**1.28**	**0.0184**	**0.0137**	**0.0053**	**0.2787**	0.257	3.9572	3.9561	20.052	3.956	3.9563	0.0334	3.1470
GPT-4v	1.56	0.0244	0.0383	0.0069	0.9664	**0.207**	**3.0218**	**3.0214**	**15.1194**	**3.0215**	3.0215	**0.0251**	**2.5028**
JanusPro 7B	1.992	0.0892	0.0782	0.0098	0.389	0.108	5.208	4.592	17.229	4.092	**2.981**	0.0782	3.0705
BLIP-2	2.992	0.064	0.192	0.023	1.102	0.942	9.28	19.029	73.923	9.284	7.128	0.103	10.3385
CogVLM	6.9543	0.166	0.112	0.196	1.644	2.912	16.482	19.597	98.439	18.502	22.844	0.0919	15.6617
mPlugOWL2	8.823	1.016	1.011	1.489	1.556	2.746	14.191	22.711	88.797	15.687	21.826	0.126	14.9983
llava	10.064	0.362	0.221	0.092	1.76	3.014	21.824	65.123	80.81	28.139	30.044	0.426	20.1566
llama-adapter	15.961	1.129	1.071	0.902	2.65	4.741	25.978	27.412	93.616	37.095	26.511	1.019	19.8404
qwen	12.988	1.203	1.102	1.091	2.813	2.988	97.619	69.377	127.947	51.125	33.136	1.018	33.5339

Table 32: ***Molecular description performance after finetuning:*** Comparison of models finetuned using LoRA on the ChEBI dataset. The best performing models are highlighted with bold text.

Models	BLEU-2 ↑	BLEU-4 ↑	ROUGE-1 ↑	ROUGE-2 ↑	ROUGE-L ↑	METEOR ↑	Average ↑
MolT5 (24)	59.40	50.80	65.40	51.00	59.40	61.40	57.90
Molca (22)	62.00	53.10	68.10	53.70	61.80	65.10	60.60
BLIP-2	59.06	58.03	58.93	58.47	58.89	58.19	58.60
CogVLM	**63.00**	**60.01**	**62.39**	**61.16**	**62.00**	**60.60**	**61.52**
mPlugOWL2	51.93	49.64	51.56	50.67	51.33	50.06	50.87
Llava 1.5 13B	60.88	58.99	60.62	59.80	60.42	59.40	60.02
Llama Adapter v2 7B	46.60	44.65	46.27	45.50	46.07	45.00	45.68
Qwen VL	52.00	50.04	51.63	50.78	51.40	50.44	51.05

description quality. All experiments use the same evaluation metrics: BLEU-2, BLEU-4, ROUGE-1, ROUGE-2, ROUGE-L, and METEOR, with average scores reported for concise comparison.

F.1 Finetuning Evaluation

Table 32 evaluates molecular description performance after LoRA finetuning on the ChEBI dataset. CogVLM achieves the best overall performance with an average score of 61.52, leading across all metrics including BLEU-4 (60.01) and ROUGE-1 (62.39). Among domain-specific baselines, Molca (60.60 average) outperforms MolT5 (57.90), while Llava 1.5 13B shows competitive results (60.02) among vision-language models. BLIP-2 delivers consistent performance across metrics (58.60 average), whereas Llama Adapter v2 7B struggles with the lowest scores (45.68 average). The results demonstrate that vision-language models can effectively generate molecular descriptions through parameter-efficient finetuning, with CogVLM showing particular strength in text generation tasks.

F.2 Zero-Shot Evaluation

We conducted experiments evaluating molecular description capabilities in a zero-shot setting. GPT-4o demonstrates superior performance across all metrics, achieving the highest average score of 27.541

(Table 33). GPT-4v follows with a notable performance gap but consistent profile (24.556 average). Among the remaining models, JanusPro and Llava 1.5 13B form the second tier (19.642 and 18.140 average, respectively), followed by CogVLM and mPlugOWL2 showing comparable capabilities (16.158 and 16.004). BLIP-2 and Qwen VL deliver similar mid-to-low range performance, while Llama Adapter v2 7B struggles significantly with this task (10.048 average). These results suggest that general-purpose models with extensive pretraining currently maintain substantial advantages in zero-shot molecular understanding and description tasks.

Table 33: ***Molecular description performance in zero-shot setting:*** Comparison of models evaluated on molecular description task without finetuning. The best performing models are highlighted with bold text.

Models	BLEU-2 ↑	BLEU-4 ↑	ROUGE-1 ↑	ROUGE-2 ↑	ROUGE-L ↑	METEOR ↑	Average ↑
GPT-4o	**27.853**	**25.491**	**29.376**	**26.184**	**28.729**	**27.615**	**27.541**
GPT-4v	24.837	23.492	25.876	24.184	25.329	23.615	24.556
BLIP-2	12.610	11.780	12.480	12.150	12.390	11.940	12.225
CogVLM	16.753	15.292	16.876	15.684	16.529	15.815	16.158
mPlugOWL2	16.621	15.220	16.432	15.837	16.286	15.629	16.004
Llava 1.5 13B	18.662	17.544	18.470	18.047	18.340	17.774	18.140
Llama Adapter v2 7B	10.332	9.723	10.235	9.985	10.173	9.841	10.048
Qwen VL	14.497	13.634	14.360	14.008	14.267	13.802	14.095
JanusPro	19.753	18.492	20.876	19.284	20.529	18.915	19.642

F.3 Effect of ICL Examples

Tables 34 and 35 demonstrate significant performance improvements when increasing from 2-shot to 4-shot learning. GPT-4v leads in the 2-shot setting with the highest average score of 43.400, while GPT-4o achieves superior results in the 4-shot setting with an average of 59.727, showing a remarkable improvement of 17.6 percentage points over its 2-shot performance. All models exhibit consistent gains when provided with additional examples, with larger models generally demonstrating better utilization of in-context examples. JanusPro maintains strong performance in both settings, while smaller models like Llama Adapter v2 7B show more modest improvements.

Table 34: ***Molecular description performance in few-shot setting (k=2):*** Comparison of models evaluated on molecular description task with 2-shot learning. The best performing models are highlighted with bold text.

Models	BLEU-2 ↑	BLEU-4 ↑	ROUGE-1 ↑	ROUGE-2 ↑	ROUGE-L ↑	METEOR ↑	Average ↑
GPT-4o	43.330	40.180	42.870	42.070	42.670	41.740	42.143
GPT-4v	**43.720**	**43.020**	**43.620**	**43.310**	**43.580**	**43.150**	**43.400**
BLIP-2	30.265	29.273	30.136	29.719	30.032	29.546	29.829
CogVLM	31.951	30.552	31.762	31.167	31.616	30.959	31.335
mPlugOWL2	26.664	24.795	26.390	25.619	26.184	25.359	25.835
Llava 1.5 13B	29.818	28.617	29.653	29.137	29.526	28.951	29.284
Llama Adapter v2 7B	21.619	20.465	21.469	21.142	21.377	20.982	21.176
Qwen VL	31.062	29.051	30.817	29.923	30.600	29.583	30.173
JanusPro	38.286	37.061	38.153	37.618	38.021	37.262	37.734

Table 35: ***Molecular description performance in few-shot setting (k=4):*** Comparison of models evaluated on molecular description task with 4-shot learning. The best performing models are highlighted with bold text.

Models	BLEU-2 ↑	BLEU-4 ↑	ROUGE-1 ↑	ROUGE-2 ↑	ROUGE-L ↑	METEOR ↑	Average ↑
GPT-4o	**61.310**	**58.690**	**60.490**	**59.220**	**60.060**	**58.590**	**59.727**
GPT-4v	55.780	54.430	55.580	54.930	55.460	54.650	55.138
BLIP-2	36.064	34.530	35.896	35.227	35.728	35.049	35.416
CogVLM	40.285	39.062	40.152	39.618	40.020	39.262	39.733
mPlugOWL2	33.774	32.299	33.614	32.969	33.454	32.792	33.150
Llava 1.5 13B	40.775	39.507	40.638	40.083	40.502	39.715	40.203
Llama Adapter v2 7B	32.063	31.075	31.947	31.456	31.857	31.298	31.616
Qwen VL	40.285	39.062	40.152	39.618	40.020	39.262	39.733
JanusPro	50.574	48.696	50.373	49.599	50.104	49.098	49.741

F.4 Chain of Thought Prompting

Table 36 illustrates the impact of Chain of Thought (CoT) prompting on molecular description tasks. GPT-4o achieves superior performance across all metrics with an average score of 61.494, slightly outperforming GPT-4v (59.549). Both models demonstrate strong capabilities when encouraged to reason step-by-step. JanusPro maintains its position as the third-best performer with an average score of 53.703, showing considerable potential among non-commercial models. The remaining models show varying degrees of effectiveness with CoT prompting, with Llava and CogVLM achieving similar performance (43.393 and 42.878, respectively). Notably, when compared to few-shot learning results, CoT prompting appears to further enhance model performance, particularly for larger models, suggesting that structured reasoning approaches are beneficial for molecular description tasks.

Table 36: ***Molecular description performance in Chain-of-thought setting:*** Comparison of models evaluated on molecular description task with Chain of Thought (CoT). The best performing models are highlighted with bold text.

Models	BLEU-2 ↑	BLEU-4 ↑	ROUGE-1 ↑	ROUGE-2 ↑	ROUGE-L ↑	METEOR ↑	Average ↑
GPT-4o	**62.162**	**60.743**	**62.406**	**61.151**	**62.006**	**60.495**	**61.494**
GPT-4v	60.242	58.784	60.026	59.324	59.897	59.022	59.549
BLIP2	38.708	37.092	38.530	37.825	38.344	37.638	38.023
CogVLM	43.467	42.187	43.324	42.745	43.182	42.361	42.878
mPlugOWL	36.475	34.883	36.303	35.606	36.130	35.416	35.802
Llava	43.996	42.668	43.849	43.249	43.702	42.892	43.393
Llama Adapter	34.628	33.561	34.503	33.972	34.405	33.802	34.145
Qwen	43.387	42.187	43.324	42.745	43.182	42.403	42.871
Janus	54.571	52.591	54.352	53.567	54.112	53.026	53.703

G Contrastive Learning for Vision Encoders

We explore two contrastive learning strategies for enhancing the vision encoder's ability to capture molecular structural information: augmentation-based and Tanimoto similarity-based approaches. This additional loss is used with LoRA finetuning. We experimente with BLIP-2 considering its better performance across all tasks. The motivation is to enable the vision encoder to learn more discriminative representations of molecular structures by leveraging either image transformations or chemical similarity relationships. The performance of these contrastive learning approaches across multiple molecular datasets is summarized in the main paper, while the detailed metrics on the molecular description task are presented in Table 37.

Table 37: ***Molecular description performance using Contrastive Learning*** Comparison of BLIP2 evaluated on molecular description task with Augmentation (Aug) and Tanimoto Augmentation (T-Aug).

Models	BLEU-2 ↑	BLEU-4 ↑	ROUGE-1 ↑	ROUGE-2 ↑	ROUGE-L ↑	METEOR ↑	Average ↑
Lora	59.062	58.034	58.933	58.466	58.889	58.185	58.595
Aug	61.530	60.307	61.398	60.863	61.266	60.507	60.979
T-Aug	**64.176**	**63.192**	**64.072**	**63.641**	**63.966**	**63.352**	**63.733**

G.1 Augmentation-based contrastive learning

We generate multiple views of the same molecule using a set of image transformations including rotations (at angles 45°, 90°, 135°, 180°, 225°, 270°, and 315°), vertical and horizontal flips, solarization, posterization, and auto-contrast adjustments. For each molecule image, we randomly apply two transformations to create positive pairs for contrastive learning.

Figure 10 illustrates the analysis of our augmentation-based approach through t-SNE visualizations of the visual encodings. While our main paper demonstrates the superior performance of contrastive learning using Tanimoto similarity, here we present results from image augmentation techniques for comparison. The left plot shows the visual encodings from BLIP-2 before cross-modal fusion, with clusters exhibiting significant overlap. The right plot displays the representations after cross-modal fusion, where the clusters become more distinguishable but still less defined than those achieved

with Tanimoto similarity methods discussed in the main paper. This comparative analysis confirms that Tanimoto similarity-based approaches provide better molecular structure differentiation than augmentation-based techniques alone, particularly for chemical structure representation tasks.

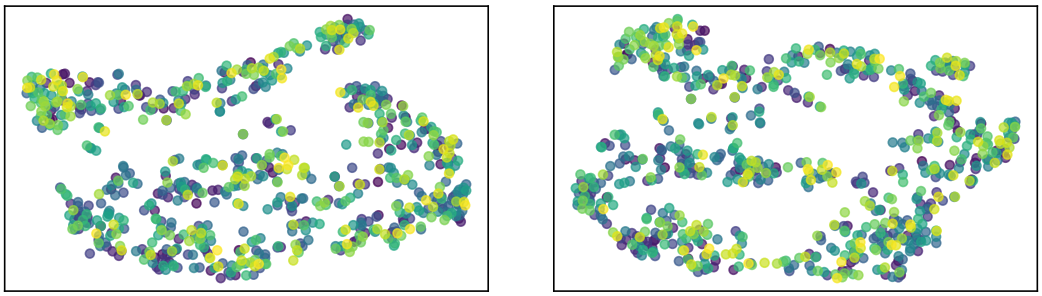

Figure 10: ***Analyzing visual features:*** The two plots show t-SNE visualizations of visual encodings of BLIP-2 before and after cross-modal fusion respectively using augmentative technique.

G.2 Tanimoto similarity-based contrastive learning

Rather than using augmented views, we leverage chemical similarity to define positive pairs. For each molecule, we identify three structurally similar molecules with Tanimoto similarity scores >0.85 to serve as positive examples. This approach ensures that the model learns from meaningful chemical relationships rather than artificial transformations.

G.3 Overall loss function

For both approaches, we implement a contrastive loss based on NT-Xent (Normalized Temperature-scaled Cross Entropy) as used in SimCLR. The fundamental principle behind this loss function is to learn discriminative molecular representations by pulling together embeddings of similar molecules (positive pairs) while pushing apart embeddings of dissimilar molecules (negative pairs) in the representation space.

The contrastive loss is mathematically defined as:

$$\mathcal{L}_{\text{contrastive}} = -\frac{1}{2N}\sum_{i=1}^{N}\log\frac{\exp(\text{sim}(z_i, z_j)/\tau)}{\sum_{k=1}^{2N}\mathbf{1}_{[k\neq i]}\exp(\text{sim}(z_i, z_k)/\tau)} \tag{1}$$

The key components of this formulation include z_i and z_j, which represent the normalized embeddings of a positive pair obtained from the vision encoder after processing molecular images. The similarity function $\text{sim}(z_i, z_j) = \frac{z_i^T z_j}{\|z_i\|\|z_j\|}$ denotes the cosine similarity between two embeddings, while $\tau = 0.5$ is the temperature parameter that controls the concentration of the distribution around positive pairs. The batch size is represented by N, resulting in $2N$ total samples when considering both elements of each positive pair, and $\mathbf{1}_{[k\neq i]}$ is an indicator function that excludes the case where $k = i$ to prevent self-comparison.

The loss function operates by computing the probability that embedding z_i is most similar to its positive counterpart z_j compared to all other embeddings in the batch. The numerator $\exp(\text{sim}(z_i, z_j)/\tau)$ represents the similarity between the positive pair, while the denominator sums over all possible negative pairs within the batch, creating a softmax-like normalization that encourages the model to distinguish between related and unrelated molecular structures.

The temperature parameter τ plays a crucial role in controlling the learning dynamics. A lower temperature such as our chosen value of 0.5 creates sharper distributions, making the model more sensitive to small differences in similarity scores and encouraging tighter clustering of positive pairs. This helps the encoder learn more discriminative features that are essential for downstream molecular property prediction across classification tasks, regression tasks, and molecular description tasks.

During training, this contrastive loss is combined with the task-specific loss for the target dataset:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{task}} + \lambda\mathcal{L}_{\text{contrastive}} \tag{2}$$

Here, $\mathcal{L}_{\text{task}}$ represents the primary loss for the specific molecular property prediction task, which varies depending on whether we are performing classification (cross-entropy loss), regression (mean squared error or mean absolute error), or molecular description (sequence generation loss). The weighting parameter λ balances the contribution of the contrastive learning objective with the task-specific objective. This combined loss function is optimized during LoRA finetuning, allowing the vision encoder to simultaneously learn task-specific features for classification, regression, and molecular description tasks while maintaining the ability to distinguish between different molecular structures through contrastive learning. The integration of contrastive learning with task-specific objectives enables the model to develop robust molecular representations that generalize well across diverse downstream applications in molecular property prediction.

H Prompt Examples

In this section, we show some prompt examples as used for various datasets. We have included some example ICL prompts specific to some of the dataset (Figure 11, 14, 12, 15,13). For regression tasks, we demonstrate an example using the ESOL dataset for solubility prediction (Figure 17). With ICL k=0 (different from zero-shot), we have all other section as shown in the prompt, however the example block is not used as input. The additional information as available with task instruction differentiates it from zero-shot, as models do not see 'Task instruction' in zero-shot. Prompt examples for ICL are included with SELFIES representations (Figure 16).

We include a chain-of-thought prompt example for the BBBP dataset (Figure 19), showing the step-by-step reasoning approach for classification tasks.

We provide a prompt example for molecular description tasks using the ChEBI dataset with ICL k=2 (Figure 18), demonstrating how the model generates natural language descriptions of molecular structures and properties.

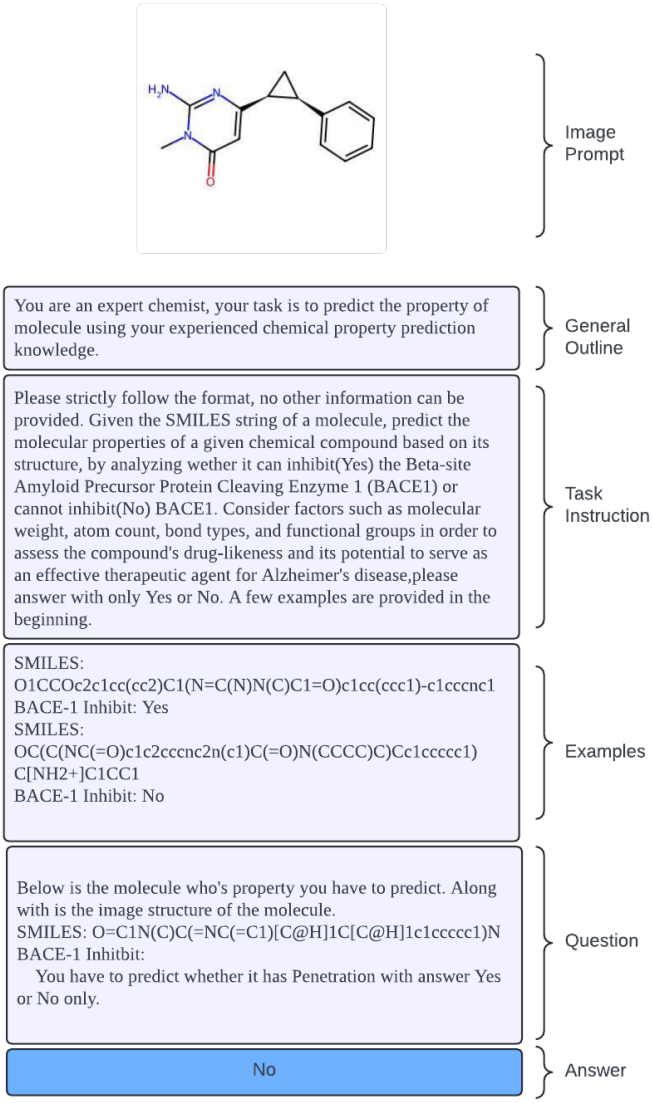

Figure 11: ***Sample prompt for BACE-V:*** A general outline is provided at first followed by set of instructions to be more specific about the task. The task is explained briefly and expected output is stated. In our case it should be Yes/No. This includes ICL examples with k=2 (No of samples). With this the main question is asked. The chemical compound who's property is to be expected is represented in its molecular structure created using RDKIT which goes along with the text input.

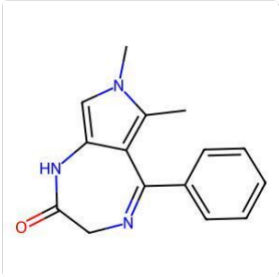

Image Prompt

You are an expert chemist, your task is to predict the property of molecule using your experienced chemical property prediction knowledge.

General Outline

Please strictly follow the format, no other information can be provided. Given the SMILES string of a molecule, the task focuses on predicting molecular properties, specifically penetration/non-penetration to the brain-blood barrier, based on the SMILES string representation of each molecule.
You will be provided with several examples molecules, each accompanied by a binary label indicating whether it has penetrative property (Yes) or not (No).
Please answer with only Yes or No.

Task Instruction

SMILES: O=C1Nc2ccccc2N1C3CCN(CCOc4ccccc4)CC3
Penetration: Yes
SMILES:
CC1CN(CCN1)c2cc3N(C=C(C(O)=O)C(=O)c3c(C)c2F)C4CC4
Penetration: No

Examples

Below is the molecule who's property you have to predict. Along with is the image structure of the molecule.
SMILES:
C1=C2C(=C([N]1C)C)C(=NCC(N2)=O)C3=CC=CC=C3
Penetration:
You have to predict whether it has Penetration with answer Yes or No only.

Question

Figure 12: ***Example prompt:*** The figure presents a task designed for predicting molecular properties, specifically penetration through the blood-brain barrier (BBBP-V dataset), using the SMILES string representation. The general outline and specific instructions detail the expected binary output (Yes/No). Two example molecules are provided to illustrate the task, followed by the main question, which includes the SMILES string and structure of the target molecule, generated with RDKit.

Image Prompt

You are an expert chemist, your task is to predict the property of molecule using your experienced chemical property prediction knowledge.

General Outline

Please strictly follow the format, no other information can be provided. Given the SMILES string of a molecule, predict the molecular properties of a given chemical compound based on its structure, by analyzing wether it can inhibit(Yes) the Beta-site Amyloid Precursor Protein Cleaving Enzyme 1 (BACE1) or cannot inhibit(No) BACE1. Consider factors such as molecular weight, atom count, bond types, and functional groups in order to assess the compound's drug-likeness and its potential to serve as an effective therapeutic agent for Alzheimer's disease,please answer with only Yes or No. A few examples are provided in the beginning.

Task Instruction

SMILES:
O1CCOc2c1cc(cc2)C1(N=C(N)N(C)C1=O)c1cc(ccc1)-c1cccnc1
BACE-1 Inhibit: Yes
SMILES:
OC(C(NC(=O)c1c2cccnc2n(c1)C(=O)N(CCCC)C)Cc1ccccc1)C[NH2+]C1CC1
BACE-1 Inhibit: No

Examples

Below is the molecule who's property you have to predict. Along with is the image structure of the molecule.
SMILES: O=C1N(C)C(=NC(=C1)[C@H]1C[C@H]1c1ccccc1)N
BACE-1 Inhitbit:
You have to predict whether it has Penetration with answer Yes or No only.

Question

No

Answer

Figure 13: ***Example prompt:*** The figure outlines a task for predicting the ability of molecules to inhibit HIV replication (HIV-V dataset), based on their SMILES string representation. The general outline and specific instructions require a binary output (Yes/No). Example molecules are provided to illustrate the task, followed by the main question, which includes the SMILES string and structure of the target molecule, generated with RDKit.

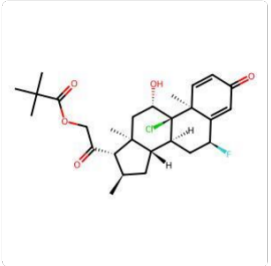

Image Prompt

You are an expert chemist, your task is to predict the property of molecule using your experienced chemical property prediction knowledge.

General Outline

Please strictly follow the format, no other information can be provided. Given the SMILES string of a molecule, the task focuses on predicting molecular properties, specifically wether a molecule is Clinically-trail-Toxic(Yes) or Not Clinically-trail-toxic (No) based on the SMILES string representation of each molecule.. The FDA-approved status will specify if the drug is approved by the FDA for clinical trials(Yes) or Not approved by the FDA for clinical trials(No).

Below is the molecule who's property you have to predict. Along with is the image structure of the molecule.

Task Instruction

SMILES:
C[C@H]1C[C@H]2[C@@H]3CCC4=CC(=O)C=C[C@@]4([C@H]3C(=O)C[C@@]2([C@]1(C(=O)CO)O)C)C
Toxic: Yes
SMILES: CC1(CC(C1)C2=NC(=C3N2C=CN=C3N)C4=CC5=C(C=C4)C=CC(=N5)C6=CC=CC=C6)O
Toxic: No

Examples

Below is the molecule who's property you have to predict. Along with is the image structure of the molecule.
SMILES: CCOC(=O)C(C(C)=O)=C(OC(C)=O)c1ccncc1C(=O)O
Inhibit:
You have to predict whether its Inhibit HIV with answer Yes or No only.

Question

Yes

Answer

Figure 14: ***Example prompt:*** The figure outlines a task for predicting whether molecules are clinically trial-toxic (ClinTox-V dataset), using their SMILES string representation. The general outline and specific instructions require a binary output (Yes/No) to indicate if the molecule is approved by the FDA for clinical trials. Example molecules are provided to illustrate the task, followed by the main question, which includes the SMILES string and structure of the target molecule, generated with RDKit.

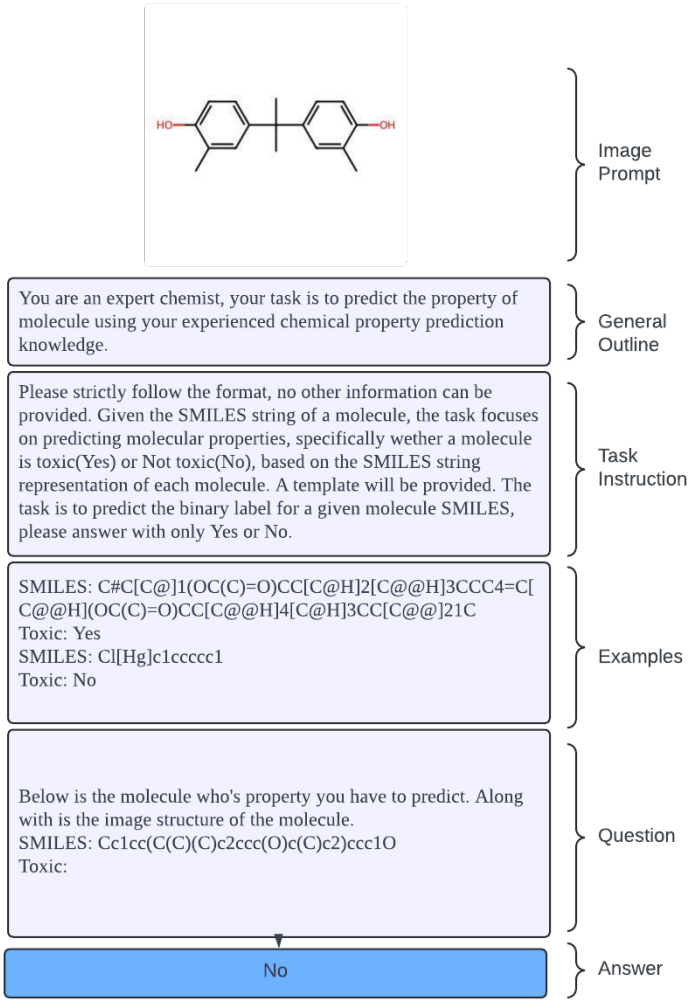

Figure 15: ***Example prompt:*** The figure outlines a task for predicting the toxicity of molecules based on their SMILES string representation, specifically in the context of the Tox21 dataset. The general outline and specific instructions require a binary output (Yes/No) to indicate the molecule's toxicity. Example molecules are provided to illustrate the task, followed by the main question, which includes the SMILES string and structure of the target molecule, generated with RDKit.

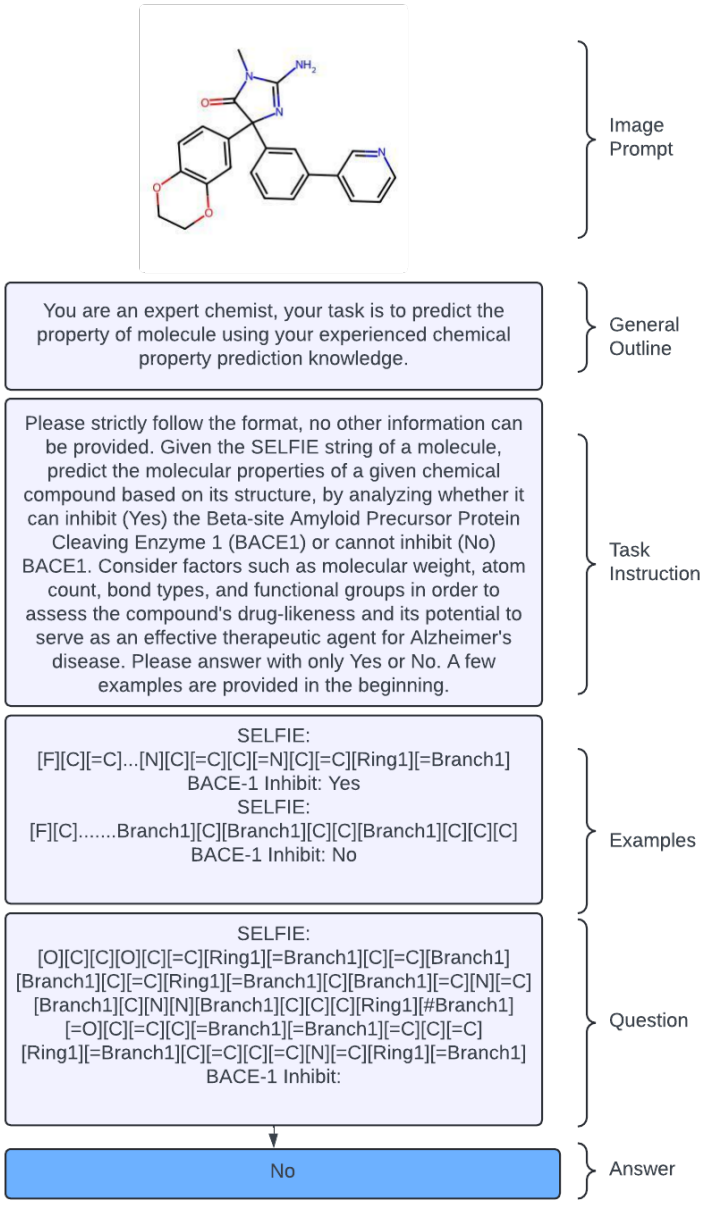

Figure 16: ***Example prompt:*** The figure outlines a task for predicting the ability of molecules for BACE-Inhibit (BACE-V dataset), using their SELFIES string representation. The general outline and specific instructions require a binary output (Yes/No). Example molecules are provided to illustrate the task, followed by the main question, which includes the SELFIES string and structure of the target molecule, generated with RDKit.

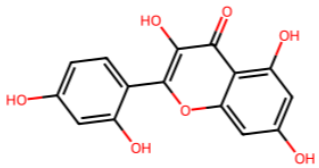

Image Prompt

As an expert chemist specializing in molecular property prediction, your task is to accurately estimate the measured log solubility in mols per litre for various compounds.

General Outline

Please strictly follow the format, no other information can be provided. Given the Smiles string of a molecule, predict the molecular properties of a given chemical compound based on its structure. You have extensive knowledge of chemical structures, solubility principles, and structure-property relationships. Consider the following information about the molecule: its molecular weight is 302.23800000000006 g/mol, and it has 5 H-bond donors. Using your expertise, analyze the given Smiles representation of the molecule, considering factors such as polarity, molecular weight, H-bond donors, and other functional groups that influence solubility. Based on this analysis, provide your best prediction of the measured log solubility in mols per litre for the following Smiles string: Oc1ccc(c(O)c1)c3oc2cc(O)cc(O)c2c(=O)c3O . You don't need any extra information for this task. Your task is to predict the answer only. The predicted output should only be a number of log solubility in mols per litre.

Task Instruction

Smiles: Nc1ccccc1ClLog Solubility: -1.52
Smiles: Oc1cccc2ncccc12 Log Solubility: -2.54

Examples

Below is the molecule who's property you have to predict. Along with is the image structure of the molecule.
Smiles: Oc1ccc(c(O)c1)c3oc2cc(O)cc(O)c2c(=O)c3O Log Solubility:

Question

-3.083

Answer

Figure 17: ***Example prompt:*** The figure outlines a task for predicting the log solubility of molecules based on their SMILES string representation, using the ESOL dataset. The general outline and specific instructions require a numerical output for log solubility in mols per litre. Two example molecules (k=2) are provided to illustrate the task, followed by the main question, which includes the SMILES string and structure of the target molecule, generated with RDKit.

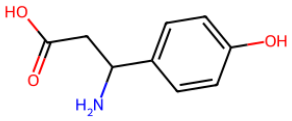

Image Prompt

You are an expert chemist, your task is to generate descriptive captions for molecules based on their chemical structure and properties.

General Outline

Given the Smiles string and molecular properties above, generate a descriptive caption for this molecule. The caption should describe the molecular structure, functional groups, potential applications, and any other relevant chemical properties. Be specific and technical in your description.

Task Instruction

Example 1:
Smiles: Nc1ccc(N)cc1
Description: The molecule is a phenylenediamine in which the amino functions are at positions 1 and 4 of the benzene nucleus. It has a role as a hapten, a dye, a reagent and an allergen.

Example 2:
Smiles: CCCCCOC(=O)c1ccccc1C(=O)O
Description: The molecule is a phthalic acid monoester obtained by formal condensation of one of the carboxy groups of phthalic acid with the hydroxy group of pentanol. It has a role as a xenobiotic metabolite, an anti-estrogen and a rat metabolite. It derives from a pentan-1-ol.

Examples

You are instructed to predict the target description caption of the molecule below. You are provided with the image of the target molecule below as a reference. Use the visual information from the image to support your analysis and reasoning, helping you generate a more accurate and comprehensive description caption of the molecule.
Target Molecule (Smiles):
NC(CC(=O)O)c1ccc(O)cc1
Description:

Question

The molecule is zwitterionic form of 3-amino-3-(4-hydroxyphenyl)propanoic acid arising from migration of a proton fron the carboxy to the amino group; major species at pH 7.3. It has a role as a bacterial metabolite. It is a tautomer of a 3-amino-3-(4-hydroxyphenyl)propanoic acid.

Answer

Figure 18: ***Example Molecular Description prompt:*** The figure outlines a task for predicting the Molecular Description based on their SMILES string representation, using the Chebi dataset. The general outline and specific instructions requires a captioning output.

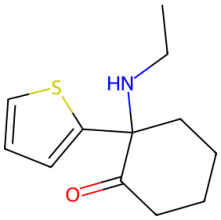

Image Prompt

You are an expert chemist, your task is to predict the property of molecule using your experienced chemical property prediction knowledge.

General Outline

You are an expert chemist tasked with predicting whether a molecule can penetrate the blood-brain barrier (BBB) based on its structure. Please follow this step-by-step reasoning process:
1. Analyze the molecular representation:
- Identify key functional groups
- Note molecular size and lipophilicity indicators
- Look for charged groups or ionizable centers
2. Consider important factors for BBB penetration:
- Molecular weight (optimal range: 400-500 Da)
- Lipophilicity (LogP between 1-4)
- Number of H bond donors (<3) and acceptors (<7)
- Polar surface area (optimal <70 Ã…Â²)
- Rotatable bonds (preferably <8)
3. Examine structural features:
- Check for presence of aromatic rings
- Identify potential P-glycoprotein substrates
- Look for CNS-unfavorable groups (acids, quaternary amines)
4. Assess drug-like properties:
- Evaluate adherence to Lipinski's Rule of Five
- Check for presence of known BBB penetration enhancers
- Consider potential metabolic stability
5. Compare to known BBB penetrators and non-penetrators:
- Analyze structural similarities with example molecules
- Note any significant structural differences
- Consider known structure-activity relationships
6. Weigh the evidence:
- List factors favoring BBB penetration
- List factors opposing BBB penetration
- Consider the overall balance of properties
7. Make a prediction:
- Based on your analysis, predict BBB penetration
- State your final prediction as either "Yes" (penetrates) or "No" (does not penetrate)
Please provide your reasoning for each step before giving your final prediction.

Task Instruction

Now, predict the BBB penetration for the following molecule:
Smiles: C1=C(SC=C1)C2(C(CCCC2)=O)NCC
Penetration:

Please follow the steps outlined above and provide your reasoning for each step before giving your final prediction of Yes or No.

Question

Yes

Answer

Figure 19: ***Example CoT prompt:*** The figure outlines a task for predicting the Brain Penetration of molecules based on their SMILES string representation, using the BBBP dataset. The general outline and specific instructions requires a binary output (Yes/No).