Generative Toolkit for Scientific Discovery

Matteo Manica\textsuperscript{1}, Jannis Born\textsuperscript{1}, Joris Cadow\textsuperscript{1}, Dimitrios Christofidellis\textsuperscript{1}, Ashish Dave\textsuperscript{1}, Dean Clarke\textsuperscript{1}, Yves Gaetan Nana Teukam\textsuperscript{1}, Giorgio Giannone\textsuperscript{1}, Samuel C. Hoffman\textsuperscript{1}, Matthew Buchan\textsuperscript{1}, Vijil Chenthamarakshan\textsuperscript{1}, Timothy Donovan\textsuperscript{1}, Hsiang Han Hsu\textsuperscript{1}, Federico Zipoli\textsuperscript{1}, Oliver Schilter\textsuperscript{1}, Akihiro Kishimoto\textsuperscript{1}, Lisa Hamada\textsuperscript{1}, Inkit Padhi\textsuperscript{1}, Karl Wehden\textsuperscript{1}, Lauren McHugh\textsuperscript{1}, Alexy Khrabrov\textsuperscript{1}, Payel Das\textsuperscript{1}, Seiji Takeda\textsuperscript{1}, and John R. Smith\textsuperscript{1}

\textsuperscript{1}IBM Research

December 5, 2022

Abstract

With the growing availability of data within various scientific domains, generative models hold enormous potential to accelerate scientific discovery. They harness powerful representations learned from datasets to speed up the formulation of novel hypotheses with the potential to impact material discovery broadly. We present the Generative Toolkit for Scientific Discovery (GT4SD). This extensible open-source library enables scientists, developers, and researchers to train and use state-of-the-art generative models to accelerate scientific discovery focused on material design.

Introduction. Humanity’s progress has been characterised by a delicate balance between curiosity and creativity. Science is no exception, with its long evolution through trial and error. While remarkably successful, the scientific method can be a slow iterative process that can be inadequate when faced with critical and pressing needs, e.g., the need to swiftly develop drugs and antibiotics or design novel materials and processes to mitigate climate change effects. Indeed, it can take almost a decade to discover a new material and cost upwards of $10–$100 million. One of the most daunting challenges in materials discovery is hypothesis generation, where it is extremely challenging to identify and select novel and useful candidates in search spaces that are overwhelming in size, e.g., the chemical space for drug-like molecules is estimated to contain \(10^{33}\) structures (Polishchuk et al., 2013).

To overcome this problem, in recent years, generative models have emerged as a practical approach to designing and discovering molecules with desired properties. Generative models more efficiently and effectively navigate and
explore vast search spaces learned from data based on user-defined criteria. With a series of seminal works (Gómez-Bombarelli et al., 2018; Segler et al., 2018; Jin et al., 2018; You et al., 2018; Prykhodko et al., 2019), research has covered a wide variety of applications of generative models, including design, optimization and discovery of: sugar and dye molecules (Takeda et al., 2020), ligands for specific targets (Zhavoronkov et al., 2019; Chenthamarakshan et al., 2020; Born et al., 2021a; Hoffman et al., 2022), anti-cancer hit-like molecules (Méndez-Lucio et al., 2020; Born et al., 2021b) and antimicrobial peptides (Das et al., 2021).

At the same time, we have witnessed growing community efforts for developing software packages to evaluate and benchmark generative models and their application in material science. Initial efforts for generic frameworks implementing popular baselines and metrics such as GuacaMol (Brown et al., 2019) and Moses (Polykovskiy et al., 2020) paved the way for domain-specific generative model software that is gaining popularity in the space of drug discovery such as TDC (Huang et al., 2021).

More recently novel families of methods have been proposed. Generative Flow Networks (GFN; (Bengio et al., 2021a,b; Jain et al., 2022)), a generative model that leverages ideas from reinforcement learning to improve sample diversity, provides a non-iterative sampling mechanism for structured data over graphs. GFNs are particularly suited for molecule generation, where sample diversity is challenging. Diffusion models (DM; (Sohl-Dickstein et al., 2015; Song and Ermon, 2019; Ho et al., 2020)) are generative models that learn complex high-dimensional distributions denoising the data at multiple scales. DMs achieve impressive results in terms of sample quality and diversity for unconditional and conditional vision tasks. Recently, text-conditional diffusion models (Ramesh et al., 2022; Rombach et al., 2022; Saharia et al., 2022) have paved the way for a new age of human-machine interaction. Leveraging such advances in conditioning generative models, DMs have been used in the biological domain for molecule conformation using equivariant graph networks (Hoogeboom et al., 2022), conditioning on a 2D representation of the molecule to generate the 3D pose in space (Xu et al., 2022), for protein generation (Anand and Achim, 2022; Wu et al., 2022), and docking (Corso et al., 2022).

**Contribution.** In this landscape, there is a growing need for libraries and toolkits that can lower the barrier to using generative models. This need is becoming significantly more pressing given the growing models’ size and companion significant requirements on considerable computational resources for training them. This trend effectively excludes large parts of the scientific community from the ability to achieve significant progress. It concentrates the power on a small, privileged group of researchers in well-funded institutions, thus impeding open, collaborative, and fair science principles.

We introduce the Generative Toolkit for Scientific Discovery (GT4SD) as a remedy. This python library aims to bridge this gap by developing a framework that eases the training, running, and developing of generative models to accelerate scientific discovery Figure 1.
GT4SD offers a set of capabilities for generating novel hypotheses (inference pipelines) and for fine-tuning domain-specific generative models (training pipelines). It is designed to be compatible and inter-operable with existing popular libraries, including PyTorch (Paszke et al., 2019), PyTorch Lightning (Falcon and The PyTorch Lightning team, 2019), Hugging Face Transformers (Wolf et al., 2020b), Diffusers (von Platen et al., 2022), GuacaMol (Brown et al., 2019), Moses (Polykovskiy et al., 2020), TorchDrug (Zhu et al., 2022), GFloW Nets (Jain et al., 2022) and MoLeR (Maziarz et al., 2021). It includes a wide range of pre-trained models and applications for material design.

GT4SD provides simple interfaces to make generative models easily accessible to users who want to deploy them with just a few lines of code. The library provides an environment for researchers and students interested in applying state-of-the-art models in their scientific research, allowing them to experiment with a wide variety of pre-trained models spanning a broad spectrum of material science and drug discovery applications. Furthermore, GT4SD provides a standardised Command Line Interface (CLI) and APIs for inference and training without compromising on the ability to specify an algorithm’s finer-grained parameters.

Results. Arguably, the most considerable potential for accelerating scientific discovery lies in the field of de novo molecular design, particularly in material and drug discovery. With several (pre)clinical trials underway (Jayatunga et al., 2022), it is a matter of time until the first AI-generated drug will receive FDA approval and reach the market. In a seminal study by (Zhavoronkov et al., 2019), a deep reinforcement learning model (GENTRL) was utilized for the discovery of potent DDR1 inhibitors, a prominent protein kinase target involved in fibrosis, cancer, and other diseases (Hidalgo-Carcedo et al., 2011). Six molecules were synthesised, four were found active in a biochemical assay, and one compound (in the following called gentrl-ddr1) demonstrated favourable pharmacokinetics in mice. As an exemplary case study in molecular discovery, we consider a contrived task of adapting the hit-compound gentrl-ddr1 to a similar molecule with an improved drug-likeness (Bickerton et al., 2012). Quantitative estimate of drug-likeness (or QED) is an in-silico property of a molecule that ranges between 0 and 1 (gentrl-ddr1: 0.38) and comprises a notion of chemical aesthetics for medicinal chemistry applications. This task requires exploring the local chemical space around the hit (i.e., gentrl-ddr1) to find an optimized lead compound.

A summary of how this task can be addressed using the GT4SD is shown in Figure 2. In the first step, a rich set of pre-trained molecular generative models is accessed with the harmonised interface of the GT4SD. Two main model classes are available. The first category is represented by graph generative models, such as MoLeR (Maziarz et al., 2022) or models from the TorchDrug library, specifically a graph-convolutional policy network and a flow-based autoregressive model (GraphAF; (Shi et al., 2020)). The second model class is chemical language models (CLM), which treat molecules as text (SMILES (Weininger, 1988) or SELFIES (Krenn et al., 2020) sequences). Most of the chemical language models in the GT4SD are accessed via the libraries MOSES (Polykovskiy et al., 2020).
or GuacaMol (Brown et al., 2019); in particular a VAE (Gómez-Bombarelli et al., 2018), an adversarial autoencoder (AAE; Kadurin et al., 2017)) or an objective-reinforced GAN model (ORGAN; Guimaraes et al., 2017)). In the first step, we randomly sample molecules from the learned chemical space of each model. Assessing the Tanimoto similarity of the generated molecules to gentrl-ddr1 reveals that this approach while producing many molecules with satisfying QED, did not sufficiently reflect the similarity constraint to the seed molecule (cf. Figure 2, bottom left). This is expected because the investigated generative models are unconditional.

As a more refined approach, the GT4SD includes conditional generative models that can be primed with continuous property constraints or molecular substructures (e.g., scaffolds) such as MoLeR (Maziarz et al., 2022), REINVENT (Blaschke et al., 2020) or even with both simultaneously (Regression Transformer; Born and Manica, 2022)). The molecules obtained from those models, in particular MoLeR and RT, largely respected the similarity constraint and produced many molecules with a Tanimoto similarity > 0.5 to gentrl-ddr1. The QED constraint was best reflected by the RT, which generated models with an improved QED up to a similarity of 0.82 (cf. Figure 2, right). In a realistic discovery scenario, the molecules generated with the described recipes could be manually reviewed by medicinal chemists and selectively considered for synthesis and screening.

GT4SD structure. The GT4SD library follows a modular structure (Figure 1) where the main components are: (i) algorithms for serving models in inference mode following a standardised API; (ii) training pipelines sharing a common interface with algorithm families-specific implementations; (iii) domain-specific utilities shared across various algorithms; (iv) a property prediction interface to evaluate generated samples (currently covering small molecules and proteins); (v) frameworks implementing support for complex workflows, e.g., granular for training mixture of generative and predictive models or enzeptional for enzyme design. Besides the core components, there are sub-modules for configuration, handling the cloud object storage-based cache, and error handling at the top-level.

GT4SD inference pipelines. The API implementation underlying the inference pipelines has been designed to support various generative model types: generation, conditional generation, controlled sampling and simple prediction algorithms. All the algorithms implemented in GT4SD follow a standard contract that guarantees a standardised way to call an algorithm in inference mode. The specific algorithm interface and applications are responsible for defining implementation details and loading the model files from a cache synced with a cloud object storage hosting their versions.

GT4SD training pipelines. Training pipelines follow the same philosophy adopted in implementing the inference pipelines. A common interface allows implementing algorithm family-specific classes with an arbitrary customisable
training method that can be configured using a set of data classes. Each training pipeline is associated with a class implementing the actual training process and a triplet of configuration data classes that control arguments for: model hyper-parameters, training parameters, and data parameters.

**GT4SD CLI commands.** To ease consumption of the pipelines and models implemented in GT4SD, a series of CLI endpoints are available alongside the package: (i) `gt4sd-inference`, to inspect and run pipelines for inference; (ii) `gt4sd-trainer`, to list and configure training pipelines; (iii) `gt4sd-saving`, to persist in a local cache a model version trained via GT4SD for usage in inference mode; (iv) `gt4sd-upload`, to upload model versions trained via GT4SD on a model hub to share algorithms with other users. The CLI commands allow to implement a complete discovery workflow where, starting from a source algorithm version, users can retrain it on custom datasets and make a new algorithm version available in GT4SD.

**Discussion.** The GT4SD is the first step toward a harmonised generative modelling environment for accelerated material discovery. For the future, we plan to expand application domains (e.g., climate, weather (Ravuri et al., 2021), sustainability, geo-informatics and human mobility (Yan et al., 2017)), and integrate novel algorithms, ideally with the support of a steadily growing open-science community.

Future developments will focus on two main components: expanding model evaluation and sample properties predictions; developing an ecosystem for sharing models built on top of the functionalities exposed via the existing CLI commands for model lifecycle management. For the first aspect, we will expand the currently integrated metrics from GuacaMol and Moses and explore bias measures to better analyse performance in light of the generated examples and their properties. Regarding the sharing ecosystem, we believe GT4SD will further benefit from an intuitive application hub that facilitates distribution of pre-trained generative models (largely inspired by the Hugging Face model hub (Wolf et al., 2020a)) and enables users to easily fine-tune models on custom data for specific applications.

We anticipate GT4SD to democratise generative modelling in the material sciences and to empower the scientific community to access, evaluate, compare and refine large-scale pre-trained models across a wide range of applications.

**Data Availability.** GT4SD source code is available on GitHub: https://github.com/GT4SD/gt4sd-core. The repository also contains exemplary notebooks and examples for users, including code and data to reproduce the presented case study. The complete documentation is available at https://gt4sd.github.io/gt4sd-core/.

**Author Contributions.** All authors contributed to the design and implementation of different library components before and after its release. M.M., J.B.,
D.C., G.G., V.C., A.K., L.M., and J.R.S. contributed to writing and revising the brief communication. J.B. designed and implemented the case study.

**Competing Interests.** The authors declare no Competing Financial or Non-Financial Interests.
References

Namrata Anand and Tudor Achim. Protein structure and sequence generation with equivariant denoising diffusion probabilistic models. arXiv preprint arXiv:2205.15019, 2022.

Emmanuel Bengio, Moksh Jain, Maksym Korablyov, Doina Precup, and Yoshua Bengio. Flow network based generative models for non-iterative diverse candidate generation. Advances in Neural Information Processing Systems, 34:27381–27394, 2021a.

Yoshua Bengio, Tristan Deleu, Edward J Hu, Salem Lahlou, Mo Tiwari, and Emmanuel Bengio. Gfownet foundations. arXiv preprint arXiv:2111.09266, 2021b.

G Richard Bickerton, Gaia V Paolini, Jérémy Besnard, Sorel Muresan, and Andrew L Hopkins. Quantifying the chemical beauty of drugs. Nature chemistry, 4(2):90–98, 2012.

Thomas Blaschke, Josep Arús-Pous, Hongming Chen, Christian Margreitter, Christian Tyrchan, Ola Engkvist, Kostas Papadopoulos, and Atanas Patronov. Reinvent 2.0: an ai tool for de novo drug design. Journal of Chemical Information and Modeling, 60(12):5918–5922, 2020.

Jannis Born and Matteo Manica. Regression transformer: Concurrent conditional generation and regression by blending numerical and textual tokens. arXiv preprint arXiv:2202.01338, 2022. Spotlight talk at ICLR workshop on Machine Learning for Drug Discovery.

Jannis Born, Matteo Manica, Joris Cadow, Greta Markert, Nil Adell Mill, Modestas Filipavicius, Nikita Janakarajan, Antonio Cardinale, Teodoro Laino, and María Rodríguez Martínez. Data-driven molecular design for discovery and synthesis of novel ligands: a case study on sars-cov-2. Machine Learning: Science and Technology, 2(2):025024, 2021a.

Jannis Born, Matteo Manica, Ali Oskooei, Joris Cadow, Greta Markert, and María Rodríguez Martínez. PaccMannRL: De novo generation of hit-like anticancer molecules from transcriptomic data via reinforcement learning. iScience, 24(4):102269, 2021b.

Nathan Brown, Marco Fiscato, Marwin HS Segler, and Alain C Vaucher. Guacamol: benchmarking models for de novo molecular design. Journal of chemical information and modeling, 59(3):1096–1108, 2019.

Vijil Chenthamarakshan, Payel Das, Samuel Hoffman, Hendrik Strobelt, Inkit Padhi, Kar Wai Lim, Benjamin Hoover, Matteo Manica, Jannis Born, Teodoro Laino, et al. Cognmol: target-specific and selective drug design for covid-19 using deep generative models. Advances in Neural Information Processing Systems, 33:4320–4332, 2020.

Gabriele Corso, Hannes Stärk, Bowen Jing, Regina Barzilay, and Tommi Jaakkola. DiffDock: Diffusion steps, twists, and turns for molecular docking. arXiv preprint arXiv:2210.01776, 2022.

Payel Das, Tom Serce, Kahini Wadhawan, Inkit Padhi, Sebastian Gehrmann, Flaviu Cipcigan, Vijil Chenthamarakshan, Hendrik Strobelt, Cicero Dos Santos, Pin-Yu Chen, et al. Accelerated antimicrobial discovery via deep generative models and molecular dynamics simulations. Nature Biomedical Engineering, 5(6):613–623, 2021.
William Falcon and The PyTorch Lightning team. PyTorch Lightning, 3 2019. URL https://github.com/PyTorchLightning/pytorch-lightning.

Rafael Gómez-Bombarelli, Jennifer N Wei, David Duvenaud, José Miguel Hernández-Lobato, Benjamín Sánchez-Lengeling, Dennis Sheberla, Jorge Aguilera-Iparraguirre, Timothy D Hirzel, Ryan P Adams, and Alán Aspuru-Guzik. Automatic chemical design using a data-driven continuous representation of molecules. ACS central science, 4(2):268–276, 2018.

Gabriel Lima Guimaraes, Benjamin Sanchez-Lengeling, Carlos Outeiral, Pedro Luis Cunha Farias, and Alán Aspuru-Guzik. Objective-reinforced generative adversarial networks (organ) for sequence generation models. arXiv preprint arXiv:1705.10843, 2017.

Cristina Hidalgo-Carcedo, Steven Hooper, Shahid I Chaudhry, Peter Williamson, Kevin Harrington, Birgit Leitinger, and Erik Sahai. Collective cell migration requires suppression of actomyosin at cell–cell contacts mediated by ddr1 and the cell polarity regulators par3 and par6. Nature cell biology, 13(1):49–59, 2011.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. Advances in Neural Information Processing Systems, 33:6840–6851, 2020.

Samuel C Hoffman, Vijil Chenthamarakshan, Kahini Wadhawan, Pin-Yu Chen, and Payel Das. Optimizing molecules using efficient queries from property evaluations. Nature Machine Intelligence, 4(1):21–31, 2022.

Emiel Hoogeboom, Victor Garcia Satorras, Clément Vignac, and Max Welling. Equivariant diffusion for molecule generation in 3d. In International Conference on Machine Learning, pages 8867–8887. PMLR, 2022.

Kexin Huang, Tianfan Fu, Wenhao Gao, Yue Zhao, Yusuf Roohani, Jure Leskovec, Connor W Coley, Cao Xiao, Jimeng Sun, and Marinka Zitnik. Therapeutics data commons: Machine learning datasets and tasks for drug discovery and development. Proceedings of Neural Information Processing Systems, NeurIPS Datasets and Benchmarks, 2021.

Moksh Jain, Emmanuel Bengio, Alex Hernandez-Garcia, Jarrid Rector-Brooks, Bonaventure FP Dosou, Chanakya Ajit Ekbote, Jie Fu, Tianyu Zhang, Michael Kilgour, Dinhguai Zhang, et al. Biological sequence design with glownets. In International Conference on Machine Learning, pages 9786–9801. PMLR, 2022.

Madura KP Jayatunga, Wen Xie, Ludwig Ruder, Ulrik Schulze, and Christoph Meier. Ai in small-molecule drug discovery: a coming wave? Nat. Rev. Drug Discov, 21: 175–176, 2022.

Wengong Jin, Regina Barzilay, and Tommi Jaakkola. Junction tree variational autoencoder for molecular graph generation. In International conference on machine learning, pages 2323–2332. PMLR, 2018.

Artur Kadurin, Alexander Aliper, Andrey Kazennov, Polina Mamoshina, Quentin Vanhaelen, Kuzma Khrabrov, and Alex Zhavoronkov. The cornucopia of meaningful leads: Applying deep adversarial autoencoders for new molecule development in oncology. Oncotarget, 8(7):10883, 2017.
Mario Krenn, Florian Häse, Akshat Kumar Nigam, Pascal Friederich, and Alan Aspuru-Guzik. Self-referencing embedded strings (SELFIES): A 100% robust molecular string representation. *Machine Learning: Science and Technology*, 1(4):045024, nov 2020. doi: 10.1088/2632-2153/aba947.

Krzysztof Maziarz, Henry Jackson-Flux, Pashmina Cameron, Finton Sirockin, Nadine Schneider, Nikolaus Stiefl, Marwin Segler, and Marc Brockschmidt. Learning to extend molecular scaffolds with structural motifs. *arXiv preprint arXiv:2103.03864*, 2021.

Krzysztof Maziarz, Henry Jackson-Flux, Pashmina Cameron, Finton Sirockin, Nadine Schneider, Nikolaus Stiefl, Marwin Segler, and Marc Brockschmidt. Learning to extend molecular scaffolds with structural motifs. In *International Conference on Learning Representations, ICLR*, 2022.

Oscar Méndez-Lucio, Benoit Baillif, Djork-Arné Clevert, David Rouquié, and Joerg Wichard. De novo generation of hit-like molecules from gene expression signatures using artificial intelligence. *Nature communications*, 11(1):1–10, 2020.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d’Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019. URL https://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf.

Pavel G Polishchuk, Timur I Madzhidov, and Alexandre Varnek. Estimation of the size of drug-like chemical space based on gdb-17 data. *Journal of computer-aided molecular design*, 27(8):675–679, 2013.

Danil Polykovskiy, Alexander Zhebrak, Benjamin Sanchez-Lengeling, Sergey Golovanov, Oktai Tatanov, Stanislav Belyaev, Rauf Kurbanov, Aleksey Artamonov, Vladimir Aladinskiy, Mark Veselov, et al. Molecular sets (moses): a benchmarking platform for molecular generation models. *Frontiers in pharmacology*, 11:1931, 2020.

Oleksii Prykhodko, Simon Viet Johansson, Panagiotis-Christos Kotsias, Josep Arús-Pous, Esben Jannik Bjerrum, Ola Engkvist, and Hongming Chen. A de novo molecular generation method using latent vector based generative adversarial network. *Journal of Cheminformatics*, 11(1):1–13, 2019.

Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 2022.

Suman Ravuri, Karel Lenc, Matthew Willson, Dmitry Kangin, Remi Lam, Piotr Mirowski, Megan Fitzsimons, Maria Athanassiadou, Sheleem Kashem, Sam Madge, et al. Skillful precipitation nowcasting using deep generative models of radar. *Nature*, 597(7878):672–677, 2021.
Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 10684–10695, 2022.

Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily Denton, Seyed Kamyar Seyed Ghasemipour, Burcu Karagol Ayan, S Sara Mahdavi, Rapha Gontijo Lopes, et al. Photorealistic text-to-image diffusion models with deep language understanding. arXiv preprint arXiv:2205.11487, 2022.

Marwin HS Segler, Thierry Kogej, Christian Tyrchan, and Mark P Waller. Generating focused molecule libraries for drug discovery with recurrent neural networks. ACS central science, 4(1):120–131, 2018.

Chence Shi, Minkai Xu, Zhaocheng Zhu, Weinan Zhang, Ming Zhang, and Jian Tang. Graphaf: a flow-based autoregressive model for molecular graph generation. In International Conference on Learning Representations, ICLR, 2020.

Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In International Conference on Machine Learning, pages 2256–2265. PMLR, 2015.

Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. Advances in Neural Information Processing Systems, 32, 2019.

Seiji Takeda, Toshiyuki Hama, Hsiang-Han Hsu, Victoria A Piumova, Dmitry Zubarev, Daniel P Sanders, Jed W Pitera, Makoto Kogoh, Takumi Hongo, Yenwei Cheng, et al. Molecular inverse-design platform for material industries. In Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pages 2961–2969, 2020.

Patrick von Platen, Suraj Patil, Anton Lozhkov, Pedro Cuenca, Nathan Lambert, Kashif Rasul, Mishig Davaadorj, and Thomas Wolf. Diffusers: State-of-the-art diffusion models. https://github.com/huggingface/diffusers, 2022.

David Weininger. Smiles, a chemical language and information system. 1. introduction to methodology and encoding rules. Journal of chemical information and computer sciences, 28(1):31–36, 1988.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. Transformers: State-of-the-art natural language processing. In Proceedings of the 2020 conference on empirical methods in natural language processing: system demonstrations, pages 38–45, 2020a.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleier, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. Transformers: State-of-the-art natural language processing. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, pages 38–45, Online, October 2020b. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/2020.emnlp-demos.6.
Kevin E Wu, Kevin K Yang, Rianne van den Berg, James Y Zou, Alex X Lu, and Ava P Amini. Protein structure generation via folding diffusion. *arXiv preprint arXiv:2209.15611*, 2022.

Minkai Xu, Lantao Yu, Yang Song, Chence Shi, Stefano Ermon, and Jian Tang. Geodiff: A geometric diffusion model for molecular conformation generation. *arXiv preprint arXiv:2203.02923*, 2022.

Xiao-Yong Yan, Wen-Xu Wang, Zi-You Gao, and Ying-Cheng Lai. Universal model of individual and population mobility on diverse spatial scales. *Nature communications*, 8(1):1–9, 2017.

Jiaxuan You, Bowen Liu, Zhitao Ying, Vijay Pande, and Jure Leskovec. Graph convolutional policy network for goal-directed molecular graph generation. *Advances in neural information processing systems*, 31, 2018.

Alex Zhavoronkov, Yan A Ivanenkov, Alex Aliper, Mark S Veselov, Vladimir A Aladinskiy, Anastasiya V Aladinskaya, Victor A Terentiev, Daniil A Polykovskiy, Maksim D Kuznetsov, Arip Asadulaev, et al. Deep learning enables rapid identification of potent ddr1 kinase inhibitors. *Nature biotechnology*, 37(9):1038–1040, 2019.

Zhaocheng Zhu, Chence Shi, Zuobai Zhang, Shengchao Liu, Minghao Xu, Xinyu Yuan, Yangtian Zhang, Junkun Chen, Huiyu Cai, Jiarui Lu, et al. Torchdrug: A powerful and flexible machine learning platform for drug discovery. *arXiv preprint arXiv:2202.08320*, 2022.
List of Figures

1 **GT4SD overview.** The library implements pipelines for inference and training of generative models. In addition, GT4SD offers utilities for algorithm versioning and sharing for broader usage in the community. The standardised interface enables algorithm instantiation and run for generating samples with less than five lines of code (top, left panel). Furthermore, the CLI tools ease the run of a full discover pipeline in the terminal (top, right panel).

2 **GT4SD structure.** The library provides (bottom, from left to right) algorithms for inference, a CLI utility, target domains, a property prediction interface, interfaces and implementations of generative modelling frameworks, and training pipelines. In the blue box, we provide a sample of available frameworks and methodologies for inference algorithms.

3 **Case study using the GT4SD for molecular discovery.** Starting from a compound designed using generative models by (Zhavoronkov et al., 2019) (gentrl-ddr1), we show how GT4SD can be used to swiftly design molecules with desired properties using a battery of algorithms available in the library in two settings: unconditional (bottom left) and conditional (bottom right). The conditional models can be constrained with chemical scaffolds or conditioned on desired property values.
Figure 1: **GT4SD overview.** The library implements pipelines for inference and training of generative models. In addition, GT4SD offers utilities for algorithm versioning and sharing for broader usage in the community. The standardised interface enables algorithm instantiation and run for generating samples with less than five lines of code (top, left panel). Furthermore, the CLI tools ease the run of a full discover pipeline in the terminal (top, right panel). **GT4SD structure.** The library provides (bottom, from left to right) algorithms for inference, a CLI utility, target domains, a property prediction interface, interfaces and implementations of generative modelling frameworks, and training pipelines. In the blue box, we provide a sample of available frameworks and methodologies for inference algorithms.
This molecule was proposed by GENTRL (a deep generative model) as DDR1-inhibitor. It showed favorable pharmacokinetics in mice. For details see Zhavoronkov et al. (2019).

Step 1: Investigate the chemical space of molecular generative models
- GCPN
- Graph AF
- MoLeR

Language models
- C3D
- VAE
- AAA
- ORGAN
- GuacaMol & MOSES

Step 2: Investigate conditional generative models that can be primed with properties or substructures (scaffolds)
- Relvent
- MoLeR
- Regression Transformer

Figure 2: Case study using the GT4SD for molecular discovery. Starting from a compound designed using generative models by (Zhavoronkov et al., 2019) (gentrl-ddr1), we show how GT4SD can be used to swiftly design molecules with desired properties using a battery of algorithms available in the library in two settings: unconditional (bottom left) and conditional (bottom right). The conditional models can be constrained with chemical scaffolds or conditioned on desired property values.