Flexible Gravitational-Wave Parameter Estimation with Transformers

Annalena Kofler[*]
MPI for Intelligent Systems, Tübingen
MPI for Gravitational Physics, Potsdam

Maximilian Dax
MPI for Intelligent Systems, Tübingen
ELLIS Institute Tübingen & Tübingen AI Center

Stephen R. Green
University of Nottingham

Jonas Wildberger
ELLIS Institute Tübingen

Nihar Gupte
MPI for Gravitational Physics, Potsdam
University of Maryland

Jakob Macke
MPI for Intelligent Systems, Tübingen
University of Tübingen & Tübingen AI Center

Alessandra Buonanno
MPI for Gravitational Physics, Potsdam
University of Maryland

Bernhard Schölkopf
MPI for Intelligent Systems, Tübingen
ELLIS Institute Tübingen

Abstract

Gravitational-wave data analysis relies on accurate and efficient methods to extract physical information from noisy detector signals, yet the increasing rate and complexity of observations represent a growing challenge. Deep learning provides a powerful alternative to traditional inference, but existing neural models typically lack the flexibility to handle variations in data analysis settings, such as missing detectors or frequency ranges. We introduce a flexible transformer-based architecture paired with a training strategy that enables adaptation to diverse analysis settings at inference time. Applied to parameter estimation, we demonstrate that a *single* flexible model—called DINGO-T1—can analyze 48 gravitational-wave events from the third LIGO-Virgo-KAGRA Observing Run under a wide range of analysis configurations, and can enable systematic studies of how detector and frequency configurations impact inferred posteriors.

1 Introduction

Gravitational waves (GWs) are ripples of space-time produced by the powerful mergers of black holes or neutron stars. These waves propagate across the Universe, and we detect them on Earth using the LIGO-Virgo-KAGRA (LVK) network of observatories [1–3]. In an ideal world, each signal would be observed by all detectors, and each detector would be functioning optimally. In reality, however, data are commonly contaminated with non-Gaussian noise artifacts, and detectors are often not in observing mode. Conventional likelihood-based parameter-estimation (PE) methods handle such issues on a case-by-case basis—for example by imposing frequency cuts or excluding non-operating detectors.

For GWs, the parameters θ characterize the astrophysical source (the masses of the merging black holes, their spins, and the space-time position and orientation of the binary) and the data d are

[*]Corresponding author: annalena.kofler@tuebingen.mpg.de

Machine Learning and the Physical Sciences Workshop, NeurIPS 2025.

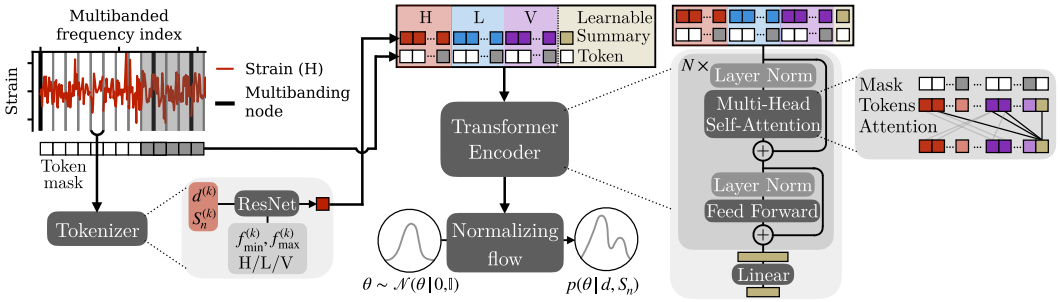

Figure 1: Architecture overview of DINGO-T1 consisting of the tokenizer, the transformer encoder and the normalizing flow.

frequency series measured in each detector. PE aims to estimate the Bayesian posterior $p(\theta|d)$ to obtain an estimate of the parameters given the observed data. Traditional approaches evaluate a likelihood that assumes additive stationary Gaussian noise, coupled with a stochastic algorithm such as nested sampling to draw posterior samples [4]. Although commonly used, this approach becomes computationally expensive in the era of large event rates. Amortized simulation-based (SBI) inference [5, 6] approaches such as DINGO [7] offer a promising solution to accelerate inference. DINGO trains NNs $q(\theta|d)$ with simulated GW data $d \sim p(d|\theta)$ to estimate $p(\theta|d)$. Once trained, $q(\theta|d_{\rm obs})$ provides rapid inference results for observed GW data $d_{\rm obs}$. This enables large-scale [8] and real-time [9] analyses, addressing limitations of conventional inference.

However, while variations in detector noise can be handled by conditioning the network on the noise curve $S_{\rm n}$, i.e., $q(\theta|d, S_{\rm n})$ [7, 10], standard neural architectures operate on a fixed input dimension, requiring the data representation to be fixed prior to training. This restricts their applicability when detectors are offline, when frequency ranges are restricted or when data gaps expected in future observatories have to be taken into account [11]. Consequently, a trained model is therefore tied to specific detector combinations and data-conditioning settings—a major barrier to widespread adoption.

To overcome this limitation, we draw inspiration from recent work on SBI with missing data [12–15] and adopt transformers [16] for GW inference. Through the self-attention mechanism, the model learns to identify correlations between strain segments, capturing global dependencies across detectors and frequencies, and to marginalize over missing data. Although transformers have been used for a range of GW tasks [17–23], previous work has not leveraged their inherent flexibility in handling variable-length or incomplete data.

Contributions. We here propose a neural architecture and training strategy that enable flexible analysis of heterogeneous GW data. The key idea is to process strain data using a transformer encoder and exploit its ability to handle input sequences of arbitrary length [16]. We partition data sets into sequences of short strain segments called tokens, and we randomly drop tokens during training to mimic frequency cuts and missing detectors. We integrate the transformer encoder into the DINGO framework for GW PE, yielding a flexible model that we call DINGO-T1 (DINGO Transformer, version 1; see Fig. 1). We demonstrate its versatility and performance through studies on simulated signals and analyses of real data. In particular, we reanalyze 48 events from the LVK third observing run (O3) [24–27], spanning 17 combinations of detectors and frequency ranges (Fig. 2a), all analyzed with a single network. Compared to baseline neural posterior estimation (NPE) networks with ResNet encoders, DINGO-T1 improves median importance sampling efficiency from 1.4% to 4.2%. Our study thus represents a step towards generalized inference models in GW astronomy, enabling flexible choice of detector and frequency settings without retraining.

2 DINGO-Transformer

For NPE [6, 28, 29], one trains a neural network $q(\theta|d)$ to estimate the Bayesian posterior $p(\theta|d)$, where d denotes the strain data measured at the detectors [30]. Data are generally of high dimension ($\gg 10^4$), so d is first mapped by an embedding network f_ϕ to a lower-dimensional representation [7]. This is then passed to a density estimator $\hat{q}_\varphi$ to model the posterior conditional on this compressed summary, $q(\theta|d) = \hat{q}_\varphi(\theta|f_\phi(d))$. The learnable parameters $\{\phi, \varphi\}$ are optimized jointly

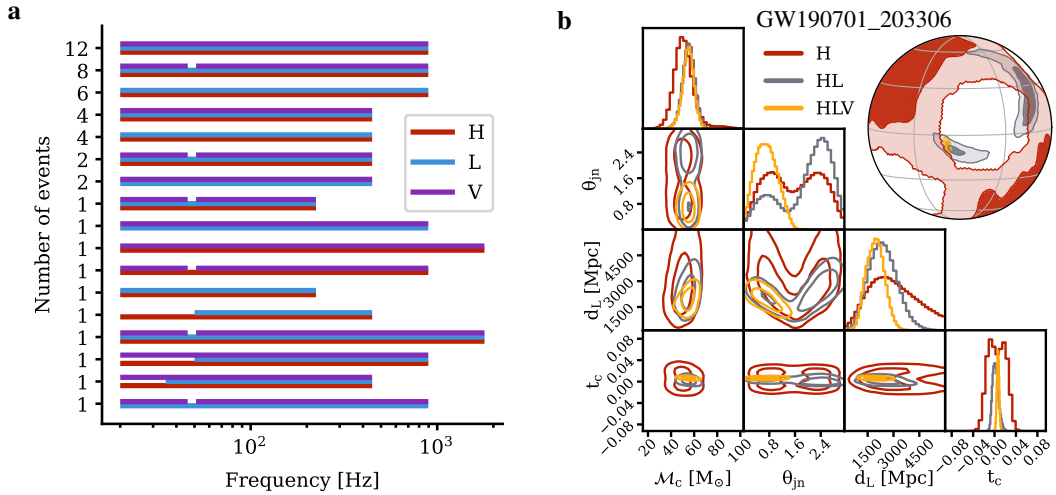

Figure 2: (a) The 48 GW events considered in this study are analyzed in 17 different data configurations in LVK catalogs [24, 25] (see also Tab. 1). Each event is analyzed with data from a subset of the three detectors HLV, and varying frequency ranges depending on data quality issues and GW source properties. The DINGO-T1 model can analyze all of these events with their respective data configurations with a single neural network. (b) Posterior distribution for GW190701_203306, showing DINGO-T1 analyses for three different detector configurations.

during training. Our core architectural change is to replace the traditional embedding network with a transformer encoder (Fig. 1), which naturally handles sequences of variable lengths. We divide the strain data sets into small segments, which a tokenizer network maps to token embeddings (Fig. 1 left). These embeddings form a sequence that the transformer encoder processes. We now describe the tokenization scheme, model architecture, and training strategy; full technical details appear in App. C.

Architecture. We represent GW data $d = \{d_I\}_{I=\mathrm{H,L,V}}$ as frequency series from the Hanford (H), Livingston (L), and Virgo (V) detectors. We use a non-uniform frequency grid with coarser resolution at higher frequencies, which captures domain knowledge about the morphology of GW signals to achieve an initial compression of the data [9, 31, 32]. To tokenize, we partition the grid into K equal-length segments with boundary nodes $(f^{(k)})_{k=0}^{K+1}$, such that each interval $\left(f^{(k)}, f^{(k+1)}\right)$ has a uniform frequency grid (Fig. 1, vertical gray lines). The data d_I and noise curve $S_{\mathrm{n},I}$ are processed identically for each detector, yielding $3K$ segments $\left(d_I^{(k)}, S_{\mathrm{n},I}^{(k)}\right)$, each containing 16 frequency bins. Each segment is then converted into a token embedding $t\left(d_I^{(k)}, S_{\mathrm{n},I}^{(k)}, f^{(k)}, f^{(k+1)}, I\right)$ by a shared tokenizer network t (see Fig. 1, bottom left).

Conditioning on frequency range and detector identity supplies the tokenizer with the necessary context for interpreting each segment [33]. The resulting token sequence is passed to a transformer encoder [16, 34, 35], which extracts information relevant for parameter estimation. We append to the sequence one additional learnable summary token so that the model has a dedicated place to store relevant information extracted across tokens [36, 37]. After the final transformer layer, we compress the summary token to a 128-dimensional feature vector, which conditions a normalizing flow with a rational-quadratic spline coupling transform [38], using the same hyperparameters as in [39].

Training. We train the DINGO-T1 model—the tokenizer, transformer encoder, and normalizing flow—end-to-end with the negative log-likelihood loss,

$$\mathcal{L} = \mathbb{E}_{p(\theta)p(S_n)p(d|\theta,S_\mathrm{n})p(m)} \left[-\log q\left(\theta | m(d), m(S_n)\right)\right]. \tag{1}$$

Here, m denotes a token mask that we sample during training and apply to (d, S_n). Tokens are masked randomly during training, enabling the model to learn to perform inference under missing information [40]. We employ a data-based approach to masking, i.e., we choose $p(m)$ to reflect realistic variations in data analysis settings. This includes dropping all tokens corresponding to certain detectors, dropping tokens to update minimum and maximum frequencies, and to apply random

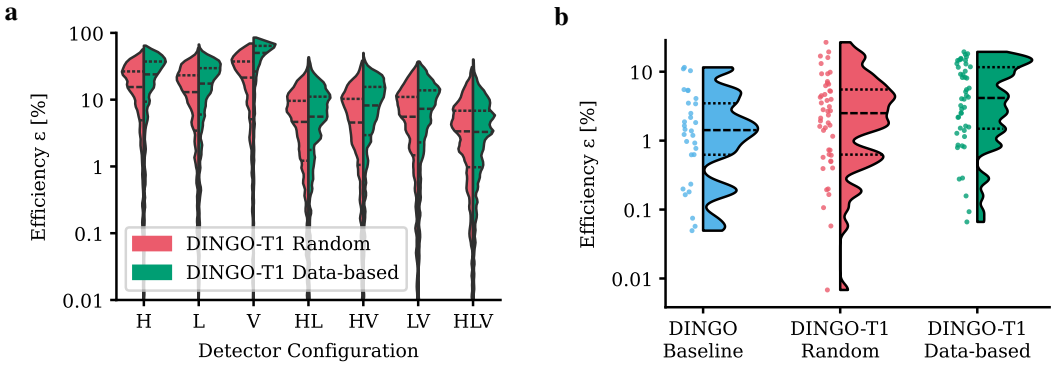

Figure 3: Sample efficiencies (↑) for (a) 1000 simulated GW signals evaluated with different detector configurations and (b) 48 real GW events. The dashed lines represents the median, the dotted lines the quartiles. We compare the DINGO-T1 model trained with data-based masking with random token masking. For the DINGO baseline in (b), we only include the 30 events for which data is available in all three detectors. Details about the performance of individual events is provided in Tab. 4.

cuts within the frequency range. We also compare to a masking strategy where tokens are dropped according to uniform draws in frequency ("random masking"). (See App. A for additional details.)

We generate $2.5 \cdot 10^7$ simulated IMRPhenomXPHM [41] waveforms with prior ranges following [39], except for the luminosity distance, for which we adopt $d_L \in [0.1, 6]$ Gpc. The waveforms are whitened by a collection of noise curves from the third observing run which we also use to condition our model on the noise level [7]. All models are trained on 8 NVIDIA-A100 GPUs using distributed data-parallel multi-GPU training which takes ∼9.5 days for the DINGO-T1 model [42, 43].

Inference. At inference time, we only include the tokens corresponding to data we wish to analyze, omitting those associated with missing information (e.g., a non-observing detector or masked frequency ranges). To validate and correct the resulting posterior, we apply importance sampling (IS) [39], reweighting samples from $q(\theta|d, S_n)$ to the exact posterior defined on the original uniform-frequency data with precise frequency ranges. The sample efficiency ϵ, computed from the importance weights, quantifies how many weighted posterior samples must be drawn to obtain a fixed number of effectively independent (unweighted) samples. Details are provided in App. F.

3 Results

First, we evaluate DINGO-T1 on 1000 *simulated* GW signals for each subset of the three detectors HLV, taking into account the full frequency range. For each signal, we generate 10^5 posterior samples, perform IS, and summarize the resulting sample efficiencies in Fig. 3a: While the model trained with random masking achieves median efficiencies of 15.7% (1-detector), 4.9% (2-detector), and 3.4% (3-detector), the efficiencies of the model trained with data-based masking are 26.9%, 6.8%, and 3.3% for one-, two-, and three-detector configurations, respectively. Efficiencies decrease as more detectors are included, consistent with expectations: posteriors constrained by more detectors are narrower and therefore more challenging to learn for normalizing flows. To show that the DINGO-T1 model is well calibrated, we provide P–P plots for each detector configuration in Fig. 5 and App. H.

Second, we evaluate DINGO-T1 on all *observed* GW events from O3 that fall within our prior, employing the specific data analysis settings from the official catalogs [24, 25] (Fig. 2a). For each event, we obtain posterior samples equivalently as above, summarize the obtained sample efficiencies in Fig. 3b and provide a detailed event list in Tab. 4. Across the 48 events, we find a median efficiency of 2.5% for the model trained with random masking and 4.2% for data-based masking. The baseline DINGO NPE model obtains a median sample efficiency of 1.4% across 30 HLV events evaluated on the full frequency range. Since the posterior distribution can change when the data is analyzed in different detectors, we evaluate events on all possible detector configurations. While such studies are computationally infeasible with standard samplers, inference times of $< 10\,\mathrm{min}$ per event make

it a reasonable analysis with DINGO-T1. We illustrate the effect on the posterior distribution at an example event in Fig. 2b.

Overall, we find that DINGO-T1 outperforms the baseline DINGO NPE model with best performance using data-based masking. For real data it should be noted that low sample efficiencies can be due to mismodeling of the signal or noise, resulting in out-of-distribution observations [39]. This does not occur in the case of simulated data.

4 Conclusions

We introduced DINGO-T1, a transformer-based model that enables flexible and fast GW parameter inference across varying data analysis settings, including detector configurations and frequency ranges. We demonstrated a realistic analysis of 48 events (including 17 analysis configurations), as well the capacity of DINGO-T1 to systematically study how detector configurations impact the inferred posterior. Thanks to improved architecture and training, we achieve a boost in median sample efficiency from 1.4% to 4.2% from an NPE baseline. While performance can be improved further by leveraging knowledge of time-translation equivariances, we place more importance on introducing a flexible model which also yields a general-purpose encoder directly enabling scaling, fine-tuning, and simple adoption in other pipelines.

Looking ahead, the transformer architecture provides a natural foundation for further extensions: (i) incorporating changes in frequency resolution across events (allowing for changes in signal duration), (ii) adapting to time- or time–frequency–domain data (similar to a vision transformer [44]), (iii) handling data gaps expected in space-based detectors such as LISA [11], or (iv) serving as a general-purpose GW data-compression backbone for downstream tasks like signal detection.

Acknowledgments and Disclosure of Funding

The authors thank Michael Pürrer and Vincent Berenz for helpful discussions. AK thanks the International Max Planck Research School for Intelligent Systems (IMPRS-IS) for support. S.R.G. is supported by a UKRI Future Leaders Fellowship (grant number MR/Y018060/1). The computational work for this Letter was performed on the Atlas cluster at the Max Planck Institute for Intelligent Systems. This research has made use of data or software obtained from the Gravitational Wave Open Science Center (gwosc.org), a service of the LIGO Scientific Collaboration, the Virgo Collaboration, and KAGRA. This material is based upon work supported by NSF's LIGO Laboratory which is a major facility fully funded by the National Science Foundation, as well as the Science and Technology Facilities Council (STFC) of the United Kingdom, the Max-Planck-Society (MPS), and the State of Niedersachsen/Germany for support of the construction of Advanced LIGO and construction and operation of the GEO600 detector. Additional support for Advanced LIGO was provided by the Australian Research Council. Virgo is funded through the European Gravitational Observatory (EGO), by the French Centre National de Recherche Scientifique (CNRS), the Italian Istituto Nazionale di Fisica Nucleare (INFN) and the Dutch Nikhef, with contributions by institutions from Belgium, Germany, Greece, Hungary, Ireland, Japan, Monaco, Poland, Portugal, Spain. KAGRA is supported by Ministry of Education, Culture, Sports, Science and Technology (MEXT), Japan Society for the Promotion of Science (JSPS) in Japan; National Research Foundation (NRF) and Ministry of Science and ICT (MSIT) in Korea; Academia Sinica (AS) and National Science and Technology Council (NSTC) in Taiwan.

References

[1] J. Aasi et al. Advanced LIGO. *Class. Quant. Grav.*, 32:074001, 2015. doi: 10.1088/0264-9381/32/7/074001.

[2] F. Acernese et al. Advanced Virgo: a second-generation interferometric gravitational wave detector. *Class. Quant. Grav.*, 32(2):024001, 2015. doi: 10.1088/0264-9381/32/2/024001.

[3] T. Akutsu et al. Overview of KAGRA: Detector design and construction history. *PTEP*, 2021 (5):05A101, 2021. doi: 10.1093/ptep/ptaa125.

[4] G. Ashton et al. BILBY: A user-friendly Bayesian inference library for gravitational-wave astronomy. *Astrophys. J. Suppl.*, 241(2):27, 2019. doi: 10.3847/1538-4365/ab06fc.

[5] K. Cranmer, J. Brehmer, and G. Louppe. The frontier of simulation-based inference. *Proceedings of the National Academy of Sciences*, 117(48):30055–30062, 2020.

[6] G. Papamakarios and I. Murray. Fast ε-free inference of simulation models with bayesian conditional density estimation. In *Advances in neural information processing systems*, 2016.

[7] M. Dax, S. R. Green, J. Gair, J. H. Macke, A. Buonanno, and B. Schölkopf. Real-Time Gravitational Wave Science with Neural Posterior Estimation. *Phys. Rev. Lett.*, 127(24):241103, 2021. doi: 10.1103/PhysRevLett.127.241103.

[8] N. Gupte et al. Evidence for eccentricity in the population of binary black holes observed by LIGO-Virgo-KAGRA, 4 2024.

[9] M. Dax et al. Real-time inference for binary neutron star mergers using machine learning. *Nature*, 639(8053):49–53, 2025. doi: 10.1038/s41586-025-08593-z.

[10] J. Wildberger et al. Adapting to noise distribution shifts in flow-based gravitational-wave inference. *Phys. Rev. D*, 107(8):084046, 2023. doi: 10.1103/PhysRevD.107.084046.

[11] O. Burke, S. Marsat, J. R. Gair, and M. L. Katz. Addressing data gaps and assessing noise mismodeling in LISA. *Phys. Rev. D*, 111(12):124053, 2025. doi: 10.1103/5jr8-k2ss.

[12] S. N. Shukla and B. M. Marlin. A survey on principles, models and methods for learning from irregularly sampled time series, 2021. URL https://arxiv.org/abs/2012.00168.

[13] Z. Wang, J. Hasenauer, and Y. Schälte. Missing data in amortized simulation-based neural posterior estimation. *PLOS Computational Biology*, 20(6):1–17, 06 2024. doi: 10.1371/journal.pcbi.1012184. URL https://doi.org/10.1371/journal.pcbi.1012184.

[14] M. Gloeckler, M. Deistler, C. D. Weilbach, F. Wood, and J. H. Macke. All-in-one simulation-based inference. In R. Salakhutdinov et al., editors, *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 15735–15766. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/gloeckler24a.html.

[15] Y. Verma, A. Bharti, and V. Garg. Robust simulation-based inference under missing data via neural processes. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=GsR3zRCRX5.

[16] A. Vaswani et al. Attention is all you need. In I. Guyon et al., editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

[17] L. Jiang and Y. Luo. Convolutional transformer for fast and accurate gravitational wave detection. In *2022 26th International Conference on Pattern Recognition (ICPR)*, pages 46–53, 2022. doi: 10.1109/ICPR56361.2022.9956104.

[18] R. Shi, Y. Zhou, T. Zhao, Z. Cao, and Z. Ren. Compact binary systems waveform generation with a generative pretrained transformer. *Phys. Rev. D*, 109:084017, Apr 2024. doi: 10.1103/PhysRevD.109.084017. URL https://link.aps.org/doi/10.1103/PhysRevD.109.084017.

[19] H. Wang, Y. Zhou, Z. Cao, Z. Guo, and Z. Ren. Waveformer: transformer-based denoising method for gravitational-wave data. *Machine Learning: Science and Technology*, 5(1): 015046, mar 2024. doi: 10.1088/2632-2153/ad2f54. URL https://dx.doi.org/10.1088/2632-2153/ad2f54.

[20] C. Chatterjee et al. Pre-trained Audio Transformer as a Foundational AI Tool for Gravitational Waves, December 2024. URL http://arxiv.org/abs/2412.20789.

[21] E. Khalouei, C. G. Sabiu, L. M. Hyung, and A. Gopakumar. External attention transformer: A robust ai model for identifying initial eccentricity signatures in binary black hole events in simulated advanced ligo data, 2025. URL https://arxiv.org/abs/2506.03634.

[22] Prasanna. M. Joshi and Reinhard Prix. Transformer networks for continuous gravitational-wave searches, 2025. URL https://arxiv.org/abs/2509.10912.

[23] L. Papalini, F. De Santi, M. Razzano, I. S. Heng, and E. Cuoco. Can Transformers help us perform parameter estimation of overlapping signals in gravitational wave detectors?, May 2025. URL http://arxiv.org/abs/2505.02773.

[24] R. Abbott, others, and The LIGO Scientific Collaboration and the Virgo Collaboration. GWTC-2.1: Deep extended catalog of compact binary coalescences observed by LIGO and Virgo during the first half of the third observing run. *Physical Review D*, 109(2):022001, January 2024. doi: 10.1103/PhysRevD.109.022001. URL https://link.aps.org/doi/10.1103/PhysRevD.109.022001.

[25] The LIGO Scientific Collaboration, the Virgo Collaboration, the KAGRA Collaboration, R. Abbott, et al. GWTC-3: Compact Binary Coalescences Observed by LIGO and Virgo During the Second Part of the Third Observing Run. *Physical Review X*, 13(4):041039, December 2023. doi: 10.1103/PhysRevX.13.041039. URL http://arxiv.org/abs/2111.03606.

[26] LIGO Scientific Collaboration and Virgo Collaboration. Gwtc-2.1: Deep extended catalog of compact binary coalescences observed by ligo and virgo during the first half of the third observing run - parameter estimation data release, May 2022. URL https://doi.org/10.5281/zenodo.6513631.

[27] LIGO Scientific Collaboration, Virgo Collaboration, and KAGRA Collaboration. Gwtc-3: Compact binary coalescences observed by ligo and virgo during the second part of the third observing run — parameter estimation data release, November 2021. URL https://doi.org/10.5281/zenodo.5546663.

[28] J.-M. Lueckmann, P. J. Gonçalves, G. Bassetto, K. Öcal, M. Nonnenmacher, and J. H. Macke. Flexible statistical inference for mechanistic models of neural dynamics. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 1289–1299, 2017.

[29] D. Greenberg, M. Nonnenmacher, and J. Macke. Automatic posterior transformation for likelihood-free inference. In *International Conference on Machine Learning*, pages 2404–2414. PMLR, 2019.

[30] Benjamin P Abbott et al. A guide to LIGO–Virgo detector noise and extraction of transient gravitational-wave signals. *Class. Quant. Grav.*, 37(5):055002, 2020. doi: 10.1088/1361-6382/ab685e.

[31] S. Vinciguerra, J. Veitch, and I. Mandel. Accelerating gravitational wave parameter estimation with multi-band template interpolation. *Class. Quant. Grav.*, 34(11):115006, 2017. doi: 10.1088/1361-6382/aa6d44.

[32] S. Morisaki. Accelerating parameter estimation of gravitational waves from compact binary coalescence using adaptive frequency resolutions. *Phys. Rev. D*, 104(4):044062, 2021. doi: 10.1103/PhysRevD.104.044062.

[33] Y. N. Dauphin, A. Fan, M. Auli, and D. Grangier. Language modeling with gated convolutional networks. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, page 933–941. JMLR, 2017.

[34] Q. Wang et al. Learning deep transformer models for machine translation. In A. Korhonen, D. Traum, and L. Màrquez, editors, *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1810–1822. Association for Computational Linguistics, July 2019. doi: 10.18653/v1/P19-1176. URL https://aclanthology.org/P19-1176/.

[35] R. Xiong et al. On layer normalization in the transformer architecture. In H. D. III and A. Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 10524–10533. PMLR, 13–18 Jul 2020. URL https://proceedings.mlr.press/v119/xiong20b.html.

[36] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In J. Burstein, C.y Doran, and T. Solorio, editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics, June 2019. doi: 10.18653/v1/N19-1423. URL https://aclanthology.org/N19-1423/.

[37] T. Darcet, M. Oquab, J. Mairal, and P. Bojanowski. Vision transformers need registers. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=2dnO3LLiJ1.

[38] C. Durkan, A. Bekasov, I. Murray, and G. Papamakarios. Neural spline flows. In H. Wallach et al., editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/file/7ac71d433f282034e088473244df8c02-Paper.pdf.

[39] M. Dax et al. Neural Importance Sampling for Rapid and Reliable Gravitational-Wave Inference. *Phys. Rev. Lett.*, 130(17):171403, 2023. doi: 10.1103/PhysRevLett.130.171403.

[40] B. Cheng, I. Misra, A. G. Schwing, A. Kirillov, and R. Girdhar. Masked-attention mask transformer for universal image segmentation. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1280–1289, 2022. doi: 10.1109/CVPR52688.2022.00135.

[41] G. Pratten et al. Computationally efficient models for the dominant and subdominant harmonic modes of precessing binary black holes. *Phys. Rev. D*, 103:104056, May 2021. doi: 10.1103/PhysRevD.103.104056. URL https://link.aps.org/doi/10.1103/PhysRevD.103.104056.

[42] I. Loshchilov and F. Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=Bkg6RiCqY7.

[43] P. Micikevicius et al. Mixed precision training. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=r1gs9JgRZ.

[44] A. Dosovitskiy et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=YicbFdNTTy.

[45] M. Dax, S. R. Green, J. Gair, M. Deistler, B. Schölkopf, and J. H. Macke. Group equivariant neural posterior estimation. In *International Conference on Learning Representations*, 11 2022.

[46] J. Sohl-Dickstein, E. Weiss, N. Maheswaranathan, and S. Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In F. Bach and D. Blei, editors, *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 2256–2265. PMLR, 2015. URL https://proceedings.mlr.press/v37/sohl-dickstein15.html.

[47] J. Ho, A. Jain, and P. Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems*, NIPS '20. Curran Associates Inc., 2020.

[48] X. Liu, C. Gong, and Q. Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. *arXiv preprint arXiv:2209.03003*, 2022.

[49] Y. Lipman, R. T. Q. Chen, H. Ben-Hamu, M. Nicke, and M. Le. Flow matching for generative modeling. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=PqvMRDCJT9t.

[50] M. S. Albergo and E. Vanden-Eijnden. Building normalizing flows with stochastic interpolants. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=li7qeBbCR1t.

[51] J. Wildberger, M. Dax, S. Buchholz, S. R. Green, J. H. Macke, and B. Schölkopf. Flow matching for scalable simulation-based inference. *NeurIPS 2023*, 12 2023.

[52] M. Leigh, D. Sengupta, G. Quétant, J. A. Raine, K. Zoch, and T. Golling. PC-JeDi: Diffusion for particle cloud generation in high energy physics. *SciPost Phys.*, 16(1):018, 2024. doi: 10.21468/SciPostPhys.16.1.018.

[53] M. Leigh, S. Klein, F. Charton, T. Golling, L. Heinrich, M. Kagan, and M. Osadchy. Is tokenization needed for masked particle modelling? In *Neurips 2024 Workshop Foundation Models for Science: Progress, Opportunities, and Challenges*, 2024. URL https://openreview.net/forum?id=F3FSFa33UI.

[54] H. W. Leung, J. Bovy, and J. S. S. Speagle. Estimating probability densities of tabular data using a transformer model combined with denoising diffusion. In *ICML 2024 Workshop on Foundation Models in the Wild*, 2024. URL https://openreview.net/forum?id=hLQnN3frL2.

[55] T. Hehn, M. Peschl, T. Orekondy, A. Behboodi, and J. Brehmer. Differentiable and learnable wireless simulation with geometric transformers. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=9TClCDZXeh.

A Data variability in GW posterior estimation

Our proposed DINGO-T1 model provides a systematic framework for handling heterogeneous data by explicitly removing masked data segments from the network input, which allows the model to flexibly accommodate the diverse data analysis settings encountered in practice. The detector configuration can change from event to event due to maintenance, instrumental issues, earthquakes, or poor data quality [24, 25]. Similarly, the analysis frequency range may be adjusted: the minimum frequency, typically $f_{\min} = 20\,\mathrm{Hz}$, can be increased to exclude non-Gaussian noise artifacts that cannot be subtracted from the data [24], as is the case for the three events highlighted in Tab. 1. The maximum frequency of each event is determined by the sampling rate f_s needed to fully resolve the signal, with $f_{\max} \propto f_s/2$ [24, 25]. In addition to such global adjustments, narrow frequency bands may also be excluded to correct for systematic calibration errors, a procedure known as "PSD notching". The idea is to set the noise curve—the power spectral density (PSD)—to an artificially large value in standard samplers, effectively preventing these frequency ranges from influencing the posterior [25]. For example, several O3b events involving the Virgo detector exhibited a narrow-band calibration error between 46 Hz and 51 Hz due to a mis-specification in the calibration models.

In summary, DINGO-T1 amortizes over key data-analysis variations—noise levels, detector setups, frequency ranges, and calibration errors—while full amortization over frequency resolution, priors, and waveform models remains future work.

B Data settings

Prior. The prior ranges follow [39], except for the luminosity distance, for which we adopt $d_L \in [0.1, 6]\ \mathrm{Gpc}$. This choice reflects that we train a single DINGO-T1 model rather than three separate models for different d_L ranges.

Frequency settings. The 48 events consistent with the chosen prior ranges were evaluated across 17 different data analysis settings, with the frequency ranges extracted from the GWTC-2.1 [26] and GWTC-3 [27] data releases and the detector configurations obtained from the corresponding catalogs [24, 25]. These configurations cover an overall frequency range of $[20,\ 1792]$ Hz, and all events are analyzed using a fixed signal duration of $T = 8\,\mathrm{s}$, leaving amortization over different frequency resolutions for future work. A complete summary of the event-specific settings is provided in Tab. 1.

Multibanded frequency domain. We employ multibanding [31, 32] to compress GW signals without loss of information before passing them to the DINGO-T1 model. In this approach, neighboring frequency bins are averaged when the signal is approximately constant, resulting in a non-uniform frequency grid with coarser resolution at high frequencies. This results in a compression method with negligible information loss. We adopt the compression scheme from [9] where the frequency resolution is decreased by a factor of 2 from one to the next multibanded frequency band. We adapt the procedure to correctly work with precessing waveforms like IMRPhenomXPHM [41] and ensure that the node position in the multibanded frequency domain corresponds to multiples of the transformer token size.

To validate the compression as loss-free, we compute the mismatch between 10^3 decimated waveforms and their interpolated counterparts under extreme conditions (minimal chirp mass and maximal time-shift). The maximal mismatch is $1.3 \cdot 10^{-3}$, comparable to the lower end of mismatches reported between IMRPhenomXPHM and highly accurate numerical relativity simulations [41].

The resulting multibanded domain defined on the frequency range $[20,\ 1810]$ Hz is used to generate a dataset of $2.5 \cdot 10^7$ waveforms for training the baseline and DINGO-T1 models. Since the multibanding nodes are defined based on the size of the token segment, we can directly partition the multibanded data into segments of fixed length that serve as tokenizer input. We adopt a token size of 16: smaller tokens (4 or 8) increase sequence length and training time without yielding noticeable improvements in model performance. Larger token sizes would result in less fine-grained control over frequency ranges at inference time. We now describe how the multibanded data segments are processed by the tokenizer within the DINGO-T1 architecture.

C Architecture details

Tokenizer The tokenizer maps each low-dimensional segment of 16 multibanded frequency bins to a token embedding vector of length $d_{\text{model}} = 1024$, using fully connected layers together with a 512-dimensional residual block. Conditional inputs—including the segment boundaries $\left(f^{(k)}, f^{(k+1)}\right)$ and the detector identity I (encoded as a one-hot vector)—are injected within the residual block via a gated linear unit [33]. This is illustrated in Fig. 1 on the left at the example of multibanded data from the Hanford detector. We also experimented with an unconditional tokenizer together with additive positional encodings as in standard language models [16]. However, such encodings are less suitable for continuous, non-uniform quantities like multibanded frequency indices, and multiple encodings would be needed to incorporate both $\left(f^{(k)}, f^{(k+1)}\right)$ and I, resulting in a large number of design choices. Over the full frequency range, we obtain 69 tokens per detector. Tokens from all detectors are concatenated together with a learnable, randomly initialized summary token [36, 37], yielding a total of $n = 69 \cdot 3 + 1 = 208$ tokens, each of size d_{model}.

Transformer Encoder The resulting token embeddings, represented as $X \in \mathbb{R}^{n \times d_{\text{model}}}$, serve as the input to the transformer encoder which captures complex correlations between the n tokens via masked self-attention:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}} + M\right)V \tag{2}$$

where the mask is defined as $M_{ij} = 0$ for allowed and $M_{ij} = -\infty$ for disallowed positions [16, 40] (see Fig. 1 on the right). The query, key, and value matrices are computed as

$$Q = X \cdot W^Q,\ K = X \cdot W^K,\ V = X \cdot W^V \tag{3}$$

with weights $W^K, W^Q, W^V \in \mathbb{R}^{d_{\text{model}} \times d_k}$. To capture multiple representation subspaces, attention is extended to h heads, whose outputs are concatenated and combined with the input X via a residual connection. Each token is passed through a feed-forward layer [16], with pre-layer normalization (LN) applied before both the self-attention and feed-forward blocks [34, 35]. We adopt the pre-LN variant instead of the original post-LN design [16] as it enables stable training without learning rate warm-up. After $N = 8$ transformer layers with $h = 16$ attention heads and feed-forward hidden size 2048, the summary token is extracted and projected to a 128-dimensional context vector using a linear layer. We found that varying the context size between 56 and 256 dimensions does not significantly affect the training loss, indicating that 128 values suffice for subsequent posterior estimation. Overall, the encoder contains $6.8 \cdot 10^7$ learnable parameters.

Normalizing flow For density estimation, we employ a conditional normalizing flow based on the rational-quadratic spline coupling transform [38], using the same architecture as in [39]. The only difference is that we train a standard NPE network instead of a group-equivariant NPE (GNPE) network [45]. The conditional normalizing flow has the same architecture for the DINGO baseline and the DINGO-T1 model, resulting in $9.2 \cdot 10^7$ learnable parameters.

Baseline We compare DINGO-T1 to a standard DINGO NPE model, which uses a residual network similar to [7] and employs layer norm instead of batch norm in the embedding network. The initial layer is seeded with components of a singular value decomposition, as in the original design, but we do not keep the initial embedding layer fixed [7] such that we can utilize the same optimizer and scheduler as the DINGO-T1 models. A summary of the model sizes is shown in Tab. 2.

D Masking strategies during training

To ensure the model can flexibly handle different data-analysis settings at inference time, it must be trained on data that reflect such variability. We therefore adopt a training strategy that mimics changing analysis conditions by removing tokens during training. Specifically, we compare two masking strategies: one uninformed by token relationships (random masking), and one that incorporates data-based structure, such as jointly masking tokens from the same detector (data-based masking), illustrated in Fig. 4.

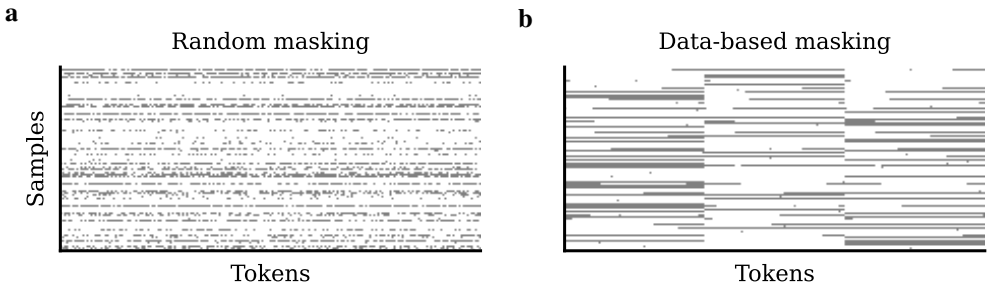

Figure 4: Comparison of (a) random and (b) data-based masking across 100 samples. Data-based masking jointly removes tokens from the same detector, yielding structured patterns, while random masking produces unstructured, scattered masks.

Random masking In the naive masking strategy, 40% of training samples are partially masked. For each masked sample, we draw the number of tokens to remove as $n_{\text{masked}} \sim \mathcal{U}[0,\ 181]$ where the upper bound allows for single-detector scenarios with additional frequency masking. The token mask m is created by randomly selecting n_{masked} tokens, without assuming any domain knowledge. This strategy thus provides unstructured masks as visible in Fig. 4a, serving as a simple baseline for testing the model's robustness to missing information.

Data-based masking Informed by the structure of GW data, we design a second masking strategy that explicitly captures realistic variations in detector configurations and frequency ranges. This procedure accounts for (1) missing detectors, (2) changes in the analysis frequency range $[f_{\min}, f_{\max}]$, and (3) narrow-band frequency removal ("PSD notching").

(1) To simulate missing detectors, we randomly decide whether to mask none (60%), one (30%), or two (10%) detectors. Among the available detectors (H/L/V), the probabilities of masking are 30%, 30%, and 40%, respectively, reflecting Virgo's larger downtime relative to the LIGO instruments. (2) To model changing frequency ranges, we apply frequency updates to 25% of training samples. For these, we mask the lower part (10%), upper part (70%), or both (20%) of the range, inspired by Fig. **??**. Since frequency updates often affect all detectors simultaneously, we apply the same mask across all detectors in 70% of these cases. New cutoff frequencies are drawn from $f_{\text{min,new}} \sim \mathcal{U}[20, 180]$ Hz and $f_{\text{max,new}} \sim \mathcal{U}[80,\ 1810]$ Hz to accommodate frequency updates required for inspiral-merger-ringdown consistency tests. In cases of $f_{\text{min,new}} > f_{\text{max,new}}$, we randomly decide to either mask the lower or upper range, depending on the probabilities given above. If a boundary falls within a token, the entire token is masked, ensuring that all affected frequency regions are consistently removed. (3) To emulate PSD notching, we mask a narrow frequency band in 10% of training samples, where the lower bound is sampled uniformly across the frequency range and the width is drawn as $\Delta f_{\text{mask}} \sim \mathcal{U}[0,\ 10]$ Hz. Overall, this data-based masking procedure closely reflects the detector and frequency conditions encountered in real gravitational-wave analyses, resulting in clearly visible boundaries between tokens from one detector in Fig. 4b. On average, 18.0% of tokens are masked during training for random masking and 25.5% for data-based masking. We experimented with masking percentages ranging from 15% to 30%, where low masking percentages resulted in lower performance in two-detector events, while applying masks to many data samples during training can lead to decreased performance in three-detector events.

E Training details

We train all models using distributed data-parallel multi-GPU training with automatic mixed precision [43] to reduce memory usage. Training is performed on 8 NVIDIA A100-SXM4-80GB GPUs (CUDA 12.1) using the AdamW [42] optimizer ($\beta_1 = 0.8$, $\beta_2 = 0.99$) with a learning rate of 0.001, weight decay of 0.005, and a `ReduceLROnPlateau` scheduler that halves the learning rate when the validation loss stagnates for 10 epochs. To ensure comparability, all models are trained with a fixed batch size of 16,384. Due to memory constraints, the DINGO-T1 models require two gradient accumulation steps per optimizer update, effectively doubling their training time. The total training duration further depends on the scheduler convergence and early-stopping conditions; detailed timings

and epoch counts are summarized in Tab. 3. Because of the long training times, we do not perform extensive hyperparameter optimization beyond a small scan over the initial learning rate, as well as β_1 and β_2 considering the first 50 epochs.

F Inference

During inference, the model has to adapt to the data analysis settings of the event of interest. Since tokens are treated as indivisible elements, any token overlapping a masked range is completely removed, which can lead to slightly larger effective exclusions than strictly necessary. Potential effects on $q(\theta|d)$ are mitigated by performing importance sampling [39] in the uniform frequency domain allowing us to adjust to the precise frequency settings. In IS, importance weights $w_i = p(\theta_i|d, S_n)p(\theta_i)/q(\theta_i|d, S_n)$ are computed for each sample $\theta_i \sim q(\theta_i|d)$. For N weighted samples, the number of effective samples from the posterior $p(\theta|d)$ corresponds to $N_{\text{eff}} = (\sum_{i=1}^{N} w_i)^2/(\sum_{i=1}^{N} w_i^2)$ and we employ the sample efficiency $\epsilon = N_{\text{eff}}/N$ as a performance metric. In practice, a specific number of effective samples is required to obtain a reliable posterior estimate, with $N_{\text{eff}} = 5,000$ defined by the LVK collaboration. Efficiencies $\epsilon > 1\%$ therefore indicate that this requirement can be reached with minimal computational cost during importance sampling; higher efficiencies (e.g., 2% vs. 15%) do not substantially reduce runtime and are therefore less critical. For $\epsilon < 1\%$, it is still possible to obtain a sufficient number of effective samples, but at an increased computational cost during IS.

In principle, it is possible to further improve performance of DINGO-T1 by incorporating additional knowledge about equivariances between data and parameters [45, 45]. However, exploiting these equivariances has certain drawbacks: (i) Group-equivariant NPE (GNPE) [45] relies on an intricate Gibbs sampling procedure at inference time to ensure that the two models jointly estimate the data standardization and posterior distribution. (ii) GNPE only provides samples from the posterior distribution, so the posterior density has to be recovered by training an unconditional normalizing flow to facilitate IS. Both points result in a complicated inference procedure making GNPE models less suitable as general purpose networks for scaling studies, fine-tuning explorations, and deployment in other pipelines. Furthermore, large inference times of roughly 1 h for 10^5 importance-sampled posterior samples on one NVIDIA-A100 GPU and 64 CPUs [39] limit exploratory interaction when changing data analysis settings. In contrast, inference times of the DINGO-T1 models are on the order of 5 - 10 min for the same number posterior samples, with initial low-latency samples being available within seconds. Inference times of DINGO-T1 are dominated by analytically reconstructing the phase ϕ_c which suffers from an inefficient implementation for IMRPhenomXPHM [39].

G Related work

Missing data in simulation-based inference Missing and irregular data present a challenge for machine learning [12] which has been addressed in the simulation-based inference community in recent works: Wang et al. [13] explored how missing values can be substituted by zero or a mean value during training. They find that imputing with a constant and providing a binary mask to the network which encodes the presence or absence of values performs most robust [13]. However, it has been shown that imputation and substituting with default values can lead to biased posterior estimates for increasing percentage of missing values [15]. Instead, it has been proposed to jointly learn an imputation and inference network employing neural processes to predict missing values before performing posterior estimation. Simformer [14] combines a transformer encoder with a diffusion model to sample arbitrary conditionals of the joint distribution over parameters and data. Missing or unstructured inputs are handled via transformer attention masks—an approach conceptually similar to the one adopted in this work. However, they do not explore problem-specific masking strategies for training and apply their approach to lower-dimensional examples.

Combining transformers with density estimation In addition to the Simformer [14], other works have combined flexible transformer encoder architectures with different density estimation techniques like diffusion models [46, 47] or flow matching [48–51]. For example, such approaches have been applied to particle cloud generation in high-energy physics [52, 53], to distributions of stellar parameters in astrophysics [54], and to inverse problems in wireless simulation [55].

Transformers in GW science Transformer encoders are been increasingly applied in GW science across a range of machine learning tasks, including noise suppression and signal reconstruction [19], signal detection [17, 18, 20], glitch classification [20], and regression of parameters [21]. More recently, transformer encoders combined with normalizing flows were used for parameter estimation of overlapping time-domain binary black hole signals in the Einstein Telescope context [23]. However, none of these approaches leverage the transformer's inherent flexibility to address the challenge of missing or incomplete data.

H Validating DINGO-T1

To ensure that the DINGO-T1 model trained with data-based masking is well calibrated, we provide P-P plots for each detector configuration. In a P-P plot, the percentile rank of the true value within its posterior marginal is computed, and the cumulative distribution function (CDF) of these ranks is visualized for each parameter. A well-calibrated model yields uniformly distributed percentiles, producing a diagonal CDF. Based on the posterior samples obtained for 1,000 injections, we construct separate P–P plots per detector configuration and include them in Fig. 5. We also report the distribution of p-values from Kolmogorov–Smirnov tests to quantify deviations from uniformity and highlight the parameters where $p < 0.05$, indicating deviations larger than expected under the uniform assumption. Overall, the DINGO-T1 models demonstrate good calibration across all detector configurations on simulated data.

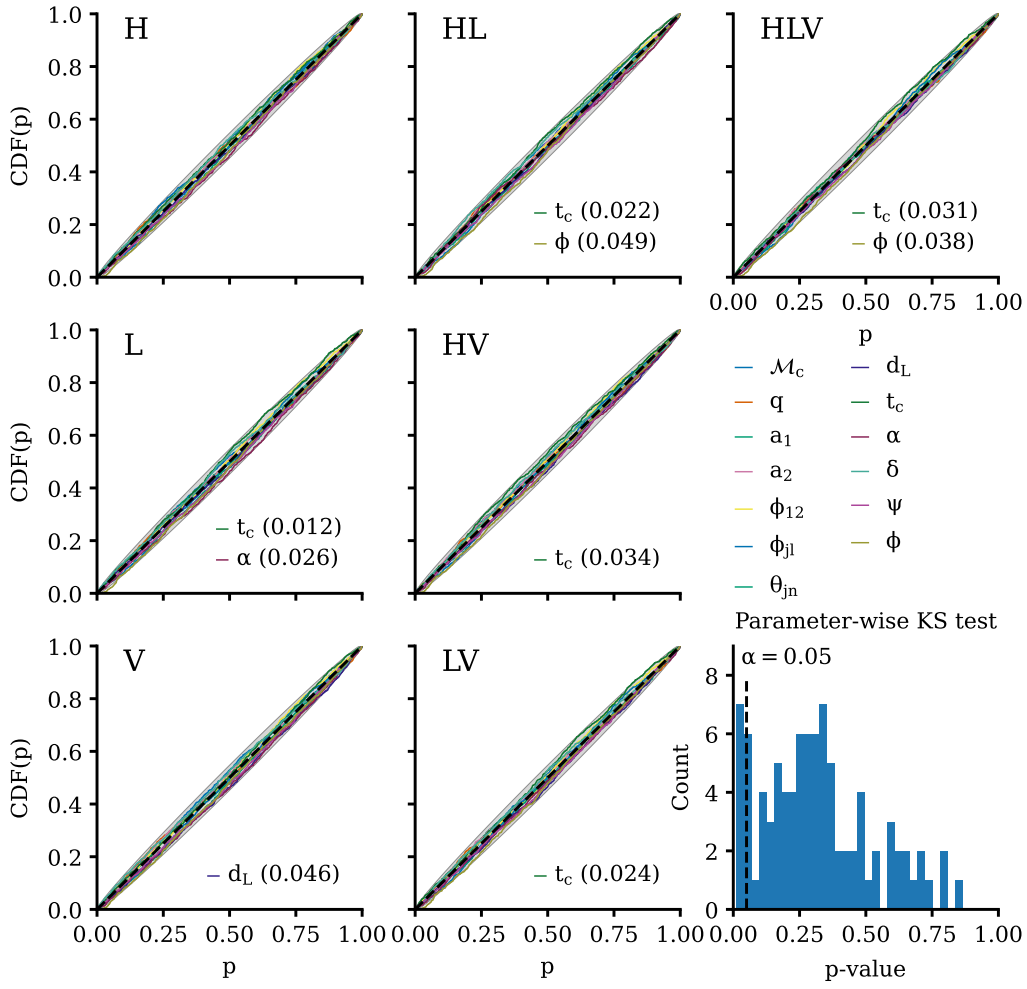

Figure 5: P-P plot for 1000 injections evaluated with different detector configurations for the DINGO-T1 model trained with data-based masking. The parameters where the Kolmogorov-Smirnov (KS) test falls below $\alpha = 0.05$ are included explicitly for each detector configuration and the overall distribution of KS-based p-values is shown in a histogram. We show the 3σ confidence intervals as a gray shaded band.

Table 1: Detector and frequency configurations for 48 events. Events with changes to $f_{\min}$ are highlighted in bold. Furthermore, we highlight events where different glitch mitigation techniques were applied to a specific detector: $*$ glitch subtraction, † update of $f_{\min}$, $\diamond$ BayesWave deglitching, and ‡ linear subtraction.

Event	Detectors	$f_{\min}$ [Hz]			$f_{\max}$[Hz]	PSD Notching (V)		Catalog
		H	L	V	H/L/V	$f_{\min}$[Hz]	$f_{\max}$[Hz]	
GW190408_181802	HLV	20	20	20	896	-	-	GWTC-2.1
GW190413_052954	HLV	20	20	20	896	-	-	GWTC-2.1
GW190413_134308$^{*\dagger L}$	HLV	20	**35**	20	448	-	-	GWTC-2.1
GW190421_213856	HL	20	20	-	448	-	-	GWTC-2.1
GW190426_190642	HLV	20	20	20	448	-	-	GWTC-2.1
GW190503_185404^{*L}	HLV	20	20	20	896	-	-	GWTC-2.1
GW190513_205428^{*L}	HLV	20	20	20	896	-	-	GWTC-2.1
GW190514_065416$^{*\dagger L}$	HL	20	**50**	-	448	-	-	GWTC-2.1
GW190517_055101	HLV	20	20	20	896	-	-	GWTC-2.1
GW190519_153544	HLV	20	20	20	448	-	-	GWTC-2.1
GW190521_074359	HL	20	20	-	224	-	-	GWTC-2.1
GW190527_092055	HL	20	20	-	896	-	-	GWTC-2.1
GW190602_175927	HLV	20	20	20	448	-	-	GWTC-2.1
GW190620_030421	LV	-	20	20	448	-	-	GWTC-2.1
GW190630_185205	LV	-	20	20	896	-	-	GWTC-2.1
GW190701_203306^{*L}	HLV	20	20	20	448	-	-	GWTC-2.1
GW190706_222641	HLV	20	20	20	896	-	-	GWTC-2.1
GW190719_215514	HL	20	20	-	896	-	-	GWTC-2.1
GW190727_060333$^{\dagger L}$	HLV	20	**50**	20	896	-	-	GWTC-2.1
GW190731_140936	HL	20	20	-	896	-	-	GWTC-2.1
GW190803_022701	HLV	20	20	20	896	-	-	GWTC-2.1
GW190828_063405	HLV	20	20	20	896	-	-	GWTC-2.1
GW190910_112807	LV	-	20	20	448	-	-	GWTC-2.1
GW190915_235702	HLV	20	20	20	896	-	-	GWTC-2.1
GW190916_200658	HLV	20	20	20	896	-	-	GWTC-2.1
GW190925_232845	HV	20	-	20	1792	-	-	GWTC-2.1
GW190926_050336	HLV	20	20	20	896	-	-	GWTC-2.1
GW190929_012149	HLV	20	20	20	896	-	-	GWTC-2.1
GW191109_010717$^{\diamond HL}$	HL	20	20	-	448	-	-	GWTC-3
GW191127_050227$^{\diamond H}$	HLV	20	20	20	896	46	51	GWTC-3
GW191204_110529	HL	20	20	-	896	-	-	GWTC-3
GW191215_223052	HLV	20	20	20	896	46	51	GWTC-3
GW191222_033537	HL	20	20	-	448	-	-	GWTC-3
GW191230_180458	HLV	20	20	20	896	46	51	GWTC-3
GW200112_155838	LV	-	20	20	896	46	51	GWTC-3
GW200128_022011	HL	20	20	-	448	-	-	GWTC-3
GW200129_065458$^{\ddagger L}$	HLV	20	20	20	896	46	51	GWTC-3
GW200208_130117	HLV	20	20	20	448	46	51	GWTC-3
GW200208_222617	HLV	20	20	20	1792	46	51	GWTC-3
GW200209_085452	HLV	20	20	20	896	46	51	GWTC-3
GW200216_220804	HLV	20	20	20	896	46	51	GWTC-3
GW200219_094415	HLV	20	20	20	896	46	51	GWTC-3
GW200220_061928	HLV	20	20	20	224	46	51	GWTC-3
GW200220_124850	HL	20	20	-	896	-	-	GWTC-3
GW200224_222234	HLV	20	20	20	448	46	51	GWTC-3
GW200302_015811	HV	20	-	20	896	46	51	GWTC-3
GW200306_093714	HL	20	20	-	896	-	-	GWTC-3
GW200311_115853	HLV	20	20	20	896	46	51	GWTC-3

Table 2: Model sizes of the compared architectures.

Method	Number of parameters		
	Embedding network	Normalizing flow	Total
DINGO Baseline	22 Mio.	92 Mio.	114 Mio.
DINGO-T1	68 Mio.	92 Mio.	160 Mio.

Table 3: Training times determined by early stopping.

Method	Epochs	Training time
DINGO Baseline	273	3d 1h
DINGO-T1 Data-based	183	9d 9h
DINGO-T1 Random	219	11d 17h

Table 4: Sample efficiencies for 48 events evaluated with 17 different frequency and detector settings for DINGO-T1. The standard DINGO network trained for HLV is evaluated on the full frequency range. Furthermore, we highlight events where different glitch mitigation techniques were applied to a specific detector: $*$ glitch subtraction, † update of $f_{\min}$, $\diamond$ BayesWave deglitching, and ‡ linear subtraction.

Event	Det.	DINGO Baseline	DINGO-T1 Random	DINGO-T1 Data-based
GW190408_181802	HLV	0.63 %	0.73 %	8.07 %
GW190413_052954	HLV	5.38 %	9.33 %	11.77 %
GW190413_134308$^{*\dagger \mathrm{L}}$	HLV	1.19 %	5.2 %	4.52 %
GW190426_190642	HLV	0.97 %	0.39 %	0.79 %
GW190503_185404$^{*\mathrm{L}}$	HLV	0.18 %	0.5 %	1.57 %
GW190513_205428$^{*\mathrm{L}}$	HLV	0.2 %	1.54 %	0.16 %
GW190517_055101	HLV	0.06 %	1.61 %	2.17 %
GW190519_153544	HLV	1.87 %	1.93 %	4.26 %
GW190602_175927	HLV	10.33 %	13.13 %	13.09 %
GW190701_203306$^{*\mathrm{L}}$	HLV	2.17 %	2.81 %	14.83 %
GW190706_222641	HLV	3.5 %	2.35 %	2.49 %
GW190727_060333$^{\dagger \mathrm{L}}$	HLV	0.77 %	1.41 %	0.84 %
GW190803_022701	HLV	11.55 %	16.86 %	16.39 %
GW190828_063405	HLV	0.92 %	4.27 %	3.99 %
GW190915_235702	HLV	3.42 %	5.29 %	8.23 %
GW190916_200658	HLV	11.0 %	19.37 %	19.41 %
GW190926_050336	HLV	1.23 %	3.52 %	1.63 %
GW190929_012149	HLV	1.64 %	1.16 %	3.04 %
GW191127_050227$^{\diamond \mathrm{H}}$	HLV	0.16 %	0.4 %	0.28 %
GW191215_223052	HLV	0.07 %	0.2 %	4.05 %
GW191230_180458	HLV	5.5 %	9.83 %	14.61 %
GW200129_065458$^{\ddagger \mathrm{L}}$	HLV	0.05 %	0.01 %	0.29 %
GW200208_130117	HLV	1.46 %	2.12 %	11.58 %
GW200208_222617	HLV	2.49 %	2.71 %	2.86 %

Event	Det.	DINGO Baseline	DINGO-T1 Random	DINGO-T1 Data-based
GW200209_085452	HLV	1.39 %	3.97 %	2.52 %
GW200216_220804	HLV	5.31 %	4.67 %	2.2 %
GW200219_094415	HLV	4.09 %	3.11 %	5.12 %
GW200220_061928	HLV	0.62 %	1.81 %	10.17 %
GW200224_222234	HLV	0.23 %	2.67 %	5.49 %
GW200311_115853	HLV	1.83 %	6.59 %	9.2 %
GW190421_213856	HL	-	9.33 %	17.8 %
GW190514_065416$^{*\dagger \mathrm{L}}$	HL	-	6.83 %	15.22 %
GW190521_074359	HL	-	0.17 %	1.28 %
GW190527_092055	HL	-	7.2 %	9.04 %
GW190719_215514	HL	-	6.26 %	14.56 %
GW190731_140936	HL	-	26.71 %	18.22 %
GW191109_010717$^{\diamond \mathrm{HL}}$	HL	-	0.19 %	1.52 %
GW191204_110529	HL	-	0.11 %	0.07 %
GW191222_033537	HL	-	16.03 %	15.61 %
GW200128_022011	HL	-	4.49 %	8.28 %
GW200220_124850	HL	-	3.83 %	12.94 %
GW200306_093714	HL	-	0.57 %	0.09 %
GW190925_232845	HV	-	0.06 %	0.86 %
GW200302_015811	HV	-	0.5 %	1.4 %
GW190620_030421	LV	-	0.63 %	0.81 %
GW190630_185205	LV	-	0.61 %	1.15 %
GW190910_112807	LV	-	0.73 %	3.22 %
GW200112_155838	LV	-	1.67 %	5.79 %