From particle clouds to tokens: building foundation models for particle physics

Joschka Birk, Anna Hallin, Gregor Kasieczka
Institute for Experimental Physics, University of Hamburg
Luruper Chaussee 149, 22761 Hamburg, Germany
joschka.birk@uni-hamburg.de

Abstract

This work presents OmniJet-α, one of the first multi-task foundation models for particle physics in the context of the Large Hadron Collider (LHC) at CERN. In contrast to natural language, particle jet data is represented by point-cloud-like objects, requiring a different type of encoding strategy to make it suitable for autoregressive generation. We introduce a comprehensive set of evaluation methods to investigate the encoding of particles into a discrete set of tokens. These methods guide us to adopt a more precise tokenization method compared to previous strategies, and we provide insights into how a rather small set of 8192 tokens can accurately represent a complex data space spanned by three continuous physical features (the momenta of the particles). Moreover, we showcase the efficacy of transfer learning between an unsupervised task (jet generation) and a common supervised task (jet tagging). This integration of disparate tasks and the successful transfer learning between them marks a significant advancement in the development of foundation models for particle physics. The code and the checkpoint of the model are available at `https://github.com/uhh-pd-ml/omnijet_alpha`.

1 Introduction

Foundation models have become the state-of-the-art approach for the most capable models in natural language processing and computer vision. Being trained on broad datasets and problems and then being able to generalize to a variety of downstream tasks and datasets [1], large-language models (LLMs) such as BERT [2], BART [3], GPT-3 [4], GPT-4 [5], and LLaMA [6] have changed the landscape of natural language processing, while models like CLIP [7] and DALL-E [8] have done the same for computer vision. The benefits of foundation models for particle physics data would be a huge leap forward. Particle physics research involves analyzing high-dimensional data from particle collisions, such as those at CERN's Large Hadron Collider (LHC). Understanding these collisions requires complex data processing and analysis pipelines, often using machine learning models that excel over classical methods [9, 10]. However, current models are tailored for specific tasks or analyses, making development and transferability challenging.

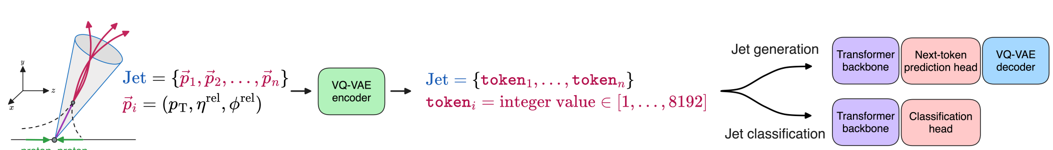

Figure 1: High-level overview of the OmniJet-α model.

Machine Learning and the Physical Sciences Workshop, NeurIPS 2024.

Foundation models on the other hand could be pre-trained on either larger simulated datasets before being fine-tuned to specific tasks or smaller datasets [11], or they could even be pre-trained on the measured data itself (of which there is an abundance). Recent efforts have demonstrated success with autoregressive generation of particle physics data [12, 13, 14]. Additionally, [15] demonstrated how a BERT-like pre-training scheme can be translated to particle physics data. Furthermore, tokenization of particle physics data was also explored in [16, 17, 14]. Our work presents one of the first multi-task foundation models in the context of *particle jets*. These particle jets are very common and important objects in particle physics, representing a collimated spray of particles that are created at particle collider experiments. As illustrated in Figure 1, we explore whether an autoregressive Generative Pre-trained Transformer (GPT) model [4] paired with a Vector Quantized Variational AutoEncoder (VQ-VAE) [18, 19, 15, 20] can be used across two distinct tasks: particle jet generation and particle jet tagging, where the latter is a common classification task in particle physics that aims to identify the type of particle that initiated a jet. Our main contributions are:

- we introduce a comprehensive set of evaluation methods to investigate the encoding of particle jets into a discrete set of tokens
- we showcase for the first time the efficacy of transfer learning between an unsupervised task (jet generation) and a common supervised task (jet tagging) in the context of particle physics, by re-using the same model architecture for both tasks, except for a small task-specific head

As this is the first model to tackle multiple tasks with jets in particle physics, it is named OMNIJET-α. Similar advancements in this domain have been made in [15, 21, 22].

All studies are performed using the JETCLASS dataset [23], which was originally introduced in [24], and contains 125M[1] jets that are extracted from proton-proton collisions, equally distributed over the following ten classes: jets initiated by gluons and quarks (q/g), top quarks (t, subdivided by their decay mode into $t \to bqq'$ and $t \to b\ell\nu$) , as well as W, Z, and H ($H \to b\bar{b}$, $H \to c\bar{c}$, $H \to gg$, $H \to 4q$, and $H \to \ell\nu qq'$) bosons. In this work, only the kinematic information per particle (p_T, ϕ, η)[2] is used while the particle mass m is approximated as zero. Furthermore, the pseudo-rapidity η and the azimuthal angle ϕ are transformed to be relative to the jet axis, i.e. $\eta^\mathrm{rel} = \eta^\mathrm{particle} - \eta^\mathrm{jet}$ and $\phi^\mathrm{rel} = \phi^\mathrm{particle} - \phi^\mathrm{jet}$, and we apply the cuts $|\eta^\mathrm{rel}| < 0.8$ and $|\phi^\mathrm{rel}| < 0.8$.

2 Particle token creation

Given that jets can have a variable number of particles, and that the particles don't have an inherent order, they are usually represented as point clouds with multiple continuous features per particle [25, 26, 9, 27, 28]. However, jets can also be represented as a sequence of tokens, which allows to use autoregressive models like GPT to generate jets. While this approach has been explored before [12, 13, 15], we take a step back to investigate the quality of the tokenization and to develop a set of quality measures to guide the choice of a suitable tokenization model. We focus on tokenization of jet constituents with a VQ-VAE [18, 19, 15, 20], where the input features are the η^rel, ϕ^rel and p_T values of the jet constituents. We compare two tokenization strategies: conditional and unconditional. In the conditional approach, a transformer is used for both encoding and decoding the constituents in a VQ-VAE, conditioned on each other. The unconditional approach uses a simple multi-layer perceptron (MLP) for encoding and decoding. We also compare VQ-VAE tokenization to a simple binning method, where input features are divided using a regular grid, with each grid cell assigned a unique token (e.g., a $10 \times 10 \times 10$ grid yields 1000 tokens). In the VQ-VAE, the input features are first encoded by an encoder, and the resulting four-dimensional latent space representation is quantized by a codebook (tokenization). This latent space representation of a jet is then decoded back to the input / physical space (token reconstruction). An important aspect of the conditional VQ-VAE is that a certain token can be reconstructed to multiple different points in physical space, depending on the other tokens in the jet. This allows the model to cover a larger space of possible jets than the unconditional VQ-VAE, which can only reconstruct a certain token to a single point in physical space.

[1] We use the default split of 100M jets for training, 5M jets for validation, and 20M jets for testing.

[2] The angle ϕ is the azimuthal angle in the transverse plane of the detector, while η is the pseudo-rapidity, a measure of the angle between the particle and the beam axis. The transverse momentum p_T is the momentum of the particle in the plane perpendicular to the beam axis.

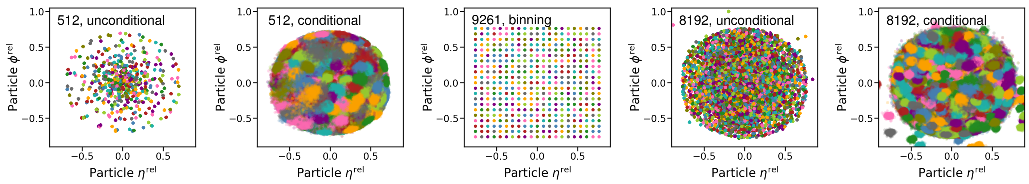

Figure 2: Visualization of reconstructed tokens in physical space (η^{rel}, ϕ^{rel}) for different tokenization approaches. Labels indicate the codebook size and the tokenization method. Unconditional and binning approaches have a single reconstruction per token. For conditional tokens, we reconstruct each token conditioned on 50 other tokens. To visualize the spread of a conditional token in physical space, we repeat this process 500 times, each time drawing 50 random tokens. Those 500 reconstructions are all drawn in the same color, resulting in a colored blob for each token.

The conditional VQ-VAE (with 8192 tokens) is trained on an NVIDIA A100 GPU for 300k training steps, taking around 20 hours of training time, whereas the unconditional VQ-VAE (with 512 tokens) converges already after a around 50k training steps, taking around 2 hours on an NVIDIA P100 GPU.

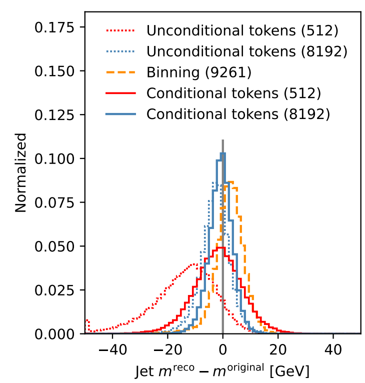

Figure 3: Difference between the jet mass of tokenized jets and the jet mass of the original jets for different tokenization approaches.

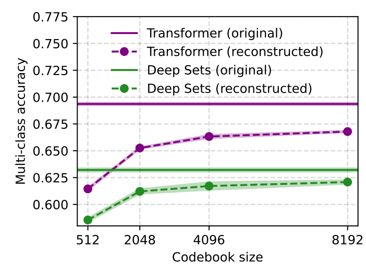

Figure 4: Token quality evaluation using a multi-class classifier, showing accuracy for different codebook sizes and classifier architectures (purple and green). Classifiers trained on original constituents provide an upper limit for accuracy.

To investigate the quality of the tokenization, we explore the following aspects: (a) the spread of the tokens in physical space on a per-particle level; (b) the quality of the tokenized jets in terms of jet-level observables; and (c) the loss of information due to the tokenization, as measured by using the reduced-resolution particles as an input to a jet classifier. A visualization of the token spread in physical space is shown in Figure 2. To assess token coverage of the physical space, we visualize reconstructed tokens as scatter plots in the $(\eta^{\mathrm{rel}}, \phi^{\mathrm{rel}})$ plane, which represents the spatial orientation of the constituent with respect to the jet axis. This is done for codebook sizes of 512 and 8192 tokens using both VQ-VAE tokenization strategies and a (21x21x21) binning method.[3] For the conditional VQ-VAE, we plot the η^{rel} and ϕ^{rel} values of each token for 500 random configurations of the remaining particles (we always reconstruct a set of 51 tokens), which results in a spread of tokens across the physical space. This spread is advantageous as it covers a large feature space with fewer tokens, unlike the unconditional VQ-VAE and binning, where each token corresponds to a single point. Multiple jet-level observables are used to evaluate the quality of the tokenized jets. Figure 3 shows the jet mass resolution for $t \to bqq'$ jets. The unconditional tokenization with a codebook size of 8192 gives the best resolution, both in terms of accuracy and spread.[4] A similar behavior can be observed for other classes, where in some cases, depending on the jet observable and the jet type, the effect is even more extreme.

To quantify the information loss due to tokenization, we train multi-class classifiers to differentiate between the ten jet types in the dataset. The classifiers are trained with both the original inputs and the inputs after tokenization and reconstruction. This comparison highlights how resolution loss affects classification performance. We use two classifier architectures: DeepSets [29, 25] (without particle interactions) and Transformer [30, 31] (with particle interactions), studying four codebook sizes ranging from 512 to 8192 tokens for the conditional tokenization approach. The resulting

[3] Chosen to match the number of tokens in VQ-VAE tokenization. The binning in p_{T} is applied to $\log(p_{\mathrm{T}})$.

[4] As expected, the resolution of the binning approach automatically leads to good resolution when the number of bins is increased to a sufficiently large number. We found that around 64 000 tokens (corresponding to a 40x40x40 grid) offer similar resolution as conditional tokenization with a codebook size of 8192.

classifier accuracy, shown in Figure 4, indicates that token quality improves with larger VQ-VAE codebook sizes, though the performance plateaus beyond 4096 tokens. Even at the largest codebook size, a performance gap remains compared to the original particles, suggesting the need for more accurate tokenization methods in future work. For the remaining studies we utilize a codebook size of 8192 with conditional tokens as this leads to the overall best performance based on our metrics.

3 Particle token generation

The core of the OmniJet-α model is a transformer backbone based on the GPT transformer decoder model from [32], using $N = 3$ GPT blocks with $n = 8$ heads in the multi-head attention blocks. During training, a *Next-token prediction head* consisting of a single linear layer is attached to the backbone. The tokens z_i are sorted by p_T in descending order before being fed to the transformer. Two additional tokens, a start token and a stop token, are added to form the sequence: $(\texttt{start\_token}, z_1, ..., z_{n-1}, z_n, \texttt{stop\_token})$. During generation, the model is provided with the start token and then auto-regressively samples the probability distribution $p\left(z_j|z_{j-1}, ..., z_1, \texttt{start\_token}\right)$ for the next token. This process is repeated until the stop token is generated or the maximum sequence length (128) is reached. The generated tokens are then decoded back to physical space using the (frozen) VQ-VAE decoder. The generative model is trained on the joint dataset of q/g and $t \to bqq'$ jets, which totals to 20M jets, for 20 epochs, taking around 20 hours of training time when trained on four NVIDIA A100 GPUs. A comparison of reconstructed JetClass jets and generated jets is shown in Figure 5 for the N-subjettiness [33] ratio τ_{32}, which is known to be a difficult observable to model. We observe that in general the model is able to match the truth level tokens well.

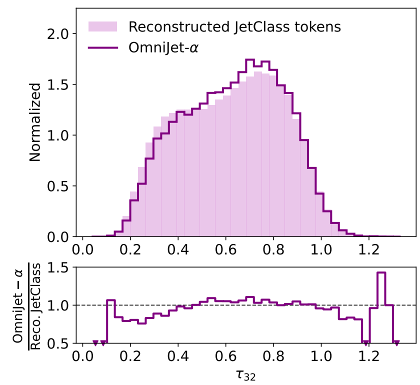

Figure 5: Comparison of the subjettiness ratio τ_{32} of generated jets from the model trained on both q/g and $t \to bqq'$ jets, to reconstructed JetClass tokens.

4 Transfer learning from generation to classification

To evaluate the ability of the model to generalize from generating jets to classifying them, we focus on the task of hadronic top quark tagging [34, 9], i.e. distinguishing $t \to bqq'$ and q/g jets on the JetClass [23] dataset. For this test, the Next-token prediction head is replaced by a *Classification head* while the transformer backbone is retained. The classification head consists of a linear layer followed by ReLU, a sum over the constituent dimension, and another linear layer with softmax activation function. We compare three training strategies: training the full architecture with randomly initialized weights (termed *from scratch*) which does not use transfer-learning and corresponds to the baseline, and two versions of fine-tuning the model obtained from the generative training. In the regular *Fine-tuning* runs, both the pre-trained backbone weights and the randomly initialized classification head weights are allowed to float in the training, while in *Fine-tuning (backbone fixed)* only the classification head is allowed to change. The results of these training runs are presented in Figure 6 as a function of the number of training examples provided to the model. We observe a significant gain in classification accuracy of both fine-tuning approaches compared to the baseline, leading to up to 17 percentage-points higher accuracy for small number of training jets, and outperforming by a few percentage-points at the highest training sample size. The difference between the two fine-tuning strategies is relatively small, with the more open training performing slightly better. Put differently, the generative pre-trained model achieves an accuracy of around 85% with 100 training examples for which the model that is trained from scratch requires more than 10 000 examples.

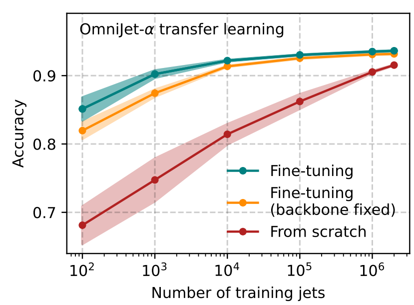

Figure 6: Performance of pre-trained and non-pre-trained models for the task of $t \to bqq'$ vs q/g jet classification.

5 Conclusion

Our investigations into strategies for effective data representations show that methods like conditional tokenization with a codebook size of 8192 help reduce information loss, which is crucial for classification and regression tasks. Moreover, OmniJet-α shows the ability to transition from unsupervised generation to supervised classification, consistently performing well compared to training from scratch, even with limited labeled data. This underscores the usefulness of foundation models in leveraging large unlabeled datasets for tasks with scarce labeled data. While our work is a step toward comprehensive foundation models, there is still room for improvement in the classification and generation performance. Further work is currently ongoing with regards to enhancing representation quality, exploring masked pre-training, scaling up architectures and training data, and expanding generalization studies. Long-term, integrating diverse datasets and embedding foundation models into community workflows are key goals.

References

[1] Rishi Bommasani et al. On the opportunities and risks of foundation models, 2022.

[2] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, 2019.

[3] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension, 2019.

[4] Tom B. Brown et al. Language Models are Few-Shot Learners, 2020.

[5] OpenAI. GPT-4 Technical Report, 2024. URL `https://arxiv.org/abs/2303.08774`.

[6] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. LLaMA: Open and Efficient Foundation Language Models, 2023.

[7] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning Transferable Visual Models From Natural Language Supervision, 2021.

[8] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical Text-Conditional Image Generation with CLIP Latents, 2022.

[9] Gregor Kasieczka et al. The machine learning landscape of top taggers. *SciPost Physics*, 7(1), July 2019. ISSN 2542-4653. doi: 10.21468/scipostphys.7.1.014. URL `http://dx.doi.org/10.21468/SciPostPhys.7.1.014`.

[10] Georgia Karagiorgi, Gregor Kasieczka, Scott Kravitz, Benjamin Nachman, and David Shih. Machine learning in the search for new fundamental physics. *Nature Rev. Phys.*, 4(6):399–412, 2022. doi: 10.1038/s42254-022-00455-1.

[11] Matthias Vigl, Nicole Hartman, and Lukas Heinrich. Finetuning Foundation Models for Joint Analysis Optimization. *Mach. Learn.: Sci. Technol. 5 025075*, 1 2024.

[12] Thorben Finke, Michael Krämer, Alexander Mück, and Jan Tönshoff. Learning the language of QCD jets with transformers. *JHEP*, 06:184, 2023. doi: 10.1007/JHEP06(2023)184.

[13] Anja Butter, Nathan Huetsch, Sofia Palacios Schweitzer, Tilman Plehn, Peter Sorrenson, and Jonas Spinner. Jet Diffusion versus JetGPT – Modern Networks for the LHC. 5 2023.

[14] Qibin Liu, Chase Shimmin, Xiulong Liu, Eli Shlizerman, Shu Li, and Shih-Chieh Hsu. Calo-VQ: Vector-Quantized Two-Stage Generative Model in Calorimeter Simulation. 5 2024.

[15] Lukas Heinrich, Tobias Golling, Michael Kagan, Samuel Klein, Matthew Leigh, Margarita Osadchy, and John Andrew Raine. Masked particle modeling on sets: Towards self-supervised high energy physics foundation models, 2024.

[16] Andris Huang, Yash Melkani, Paolo Calafiura, Alina Lazar, Daniel Thomas Murnane, Minh-Tuan Pham, and Xiangyang Ju. A Language Model for Particle Tracking. In *Connecting The Dots 2023*, 2 2024.

[17] Baran Hashemi, Nikolai Hartmann, Sahand Sharifzadeh, James Kahn, and Thomas Kuhr. Ultra-high-granularity detector simulation with intra-event aware generative adversarial network and self-supervised relational reasoning. *Nature Commun.*, 15(1):4916, 2024. doi: 10.1038/s41467-024-49104-4. [Erratum: Nature Commun. 115, 5825 (2024)].

[18] Aaron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. Neural Discrete Representation Learning, 2018.

[19] Hangbo Bao, Li Dong, Songhao Piao, and Furu Wei. BEiT: BERT Pre-Training of Image Transformers, 2022.

[20] Minyoung Huh, Brian Cheung, Pulkit Agrawal, and Phillip Isola. Straightening Out the Straight-Through Estimator: Overcoming Optimization Challenges in Vector Quantized Networks, 2023.

[21] Philip Harris, Michael Kagan, Jeffrey Krupa, Benedikt Maier, and Nathaniel Woodward. Re-Simulation-based Self-Supervised Learning for Pre-Training Foundation Models. 3 2024.

[22] Vinicius Mikuni and Benjamin Nachman. OmniLearn: A Method to Simultaneously Facilitate All Jet Physics Tasks. 4 2024.

[23] Huilin Qu, Congqiao Li, and Sitian Qian. JetClass: A Large-Scale Dataset for Deep Learning in Jet Physics, 6 2022. URL `https://doi.org/10.5281/zenodo.6619768`.

[24] Huilin Qu, Congqiao Li, and Sitian Qian. Particle Transformer for Jet Tagging. In *Proceedings of the 39th International Conference on Machine Learning*, pages 18281–18292, 2022.

[25] Patrick T. Komiske, Eric M. Metodiev, and Jesse Thaler. Energy flow networks: deep sets for particle jets. *Journal of High Energy Physics*, 2019(1), January 2019. ISSN 1029-8479. doi: 10.1007/jhep01(2019)121. URL `http://dx.doi.org/10.1007/JHEP01(2019)121`.

[26] Erik Buhmann, Gregor Kasieczka, and Jesse Thaler. EPiC-GAN: Equivariant Point Cloud Generation for Particle Jets, 2023.

[27] Erik Buhmann, Sascha Diefenbacher, Engin Eren, Frank Gaede, Gregor Kasieczka, Anatolii Korol, William Korcari, Katja Krüger, and Peter McKeown. CaloClouds: fast geometry-independent highly-granular calorimeter simulation. *JINST*, 18(11):P11025, 2023. doi: 10.1088/1748-0221/18/11/P11025.

[28] Erik Buhmann, Frank Gaede, Gregor Kasieczka, Anatolii Korol, William Korcari, Katja Krüger, and Peter McKeown. CaloClouds II: ultra-fast geometry-independent highly-granular calorimeter simulation. *JINST*, 19(04):P04020, 2024. doi: 10.1088/1748-0221/19/04/P04020.

[29] Manzil Zaheer, Satwik Kottur, Siamak Ravanbakhsh, Barnabas Poczos, Ruslan Salakhutdinov, and Alexander Smola. Deep Sets, 2018.

[30] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention Is All You Need. In *31st International Conference on Neural Information Processing Systems*, 6 2017.

[31] Sam Shleifer, Jason Weston, and Myle Ott. Normformer: Improved transformer pretraining with extra normalization, 2021.

[32] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pre-training. 2018. URL `https://cdn.openai.com/research-covers/language-unsupervised/language_understanding_paper.pdf`.

[33] Jesse Thaler and Ken Van Tilburg. Identifying Boosted Objects with N-subjettiness. *JHEP*, 03:015, 2011. doi: 10.1007/JHEP03(2011)015.

[34] Gregor Kasieczka, Tilman Plehn, Michael Russell, and Torben Schell. Deep-learning Top Taggers or The End of QCD? *JHEP*, 05:006, 2017. doi: 10.1007/JHEP05(2017)006.