Project Proposal

Background Enrichment augmented Anomaly Detection (BEAD) for new physics searches at LHC

Abhishek Kotwani
Country: India
Email: abhishek.9.kotwani@gmail.com
Github: Abhi-sheKkK
Timezone: IST (GMT + 5:30)
Resume: Abhishek_resume_v1
Mentors: Pratik Jawahar, Sukanya Sinha, Caterina Doglioni

Table of Content

1.Introduction:

1.1 About Me

I’m a **second-year** college student with an interest in **AI** and **machine learning** (AIML), beginning with **C++** in 12th grade before followed by mastering **Python** in college. My journey in **deep learning** started off with **Transformers**, where I independently implemented "**Stock Transformer**" independently with **PyTorch**.I also participated in **Hacktoberfest 2024** with 4 successfully merged pull requests. Projects like **drug innovation with VAEs** and a **reinforcement learning traffic system** for **Smart India Hackathon** cemented my expertise. Fascinated by quantum mechanics, I’m eager to explore physics at a quantum level. My experience with **Transformers, VAEs, PyTorch**, and problem-solving drives me to contribute to this GSoC project on anomaly detection in high energy physics.

1.2 Studies

I’m in my Second Year at **Veermata Jijabai Technological Institute**, Mumbai, India

Degree: Electronics Engineering (Minor in AI/ML)

Location: Mumbai, India

Relevant Coursework: Python Programming,Deep Learning,Machine Learning, Artificial intelligence.

Availability: I can contribute around **20 - 30 hrs** a week, at an average of **4-5** hrs a day. With extra time on weekends to catch up with any remaining work and documentation.

(My application would not affect my ongoing degree)

1.3 Specifications

I’m using a ASUS Vivobook with dual boot windows and linux .

OS: Ubuntu 22.04 LTS 64-bit , Windows 11

Processor: Intel i5-9300H

2.Why this project?

2.1 Motivation

As a **second-year** AIML enthusiast fascinated by **quantum mechanics**, I'm drawn to this project's fusion of **deep learning** and **high energy physics**. It tackles **anomaly detection** in **jet data** with innovative **VAEs** and **Transformers**—tools I've explored in my previous projects. This aligns with my goal to unravel quantum-level secrets using **ML**, offering a chance to impact real physics discovery.

2.2 Previous Interaction with This Project

- Created a **job submission shell script** to run all available **Normalizing Flow (NF) + ConvVAE** models for 500 epochs automatically, saving models at each 100 epochs for evaluation.
- This script was written to submit a job on CSF3.
- I also integrated a **Vanilla VAE** model into the project's repository as part of the screening test.
- Created a detailed logger report documenting the script's functionality, line-by-line explanations, and the rationale behind each design choice.

3.Research Contribution

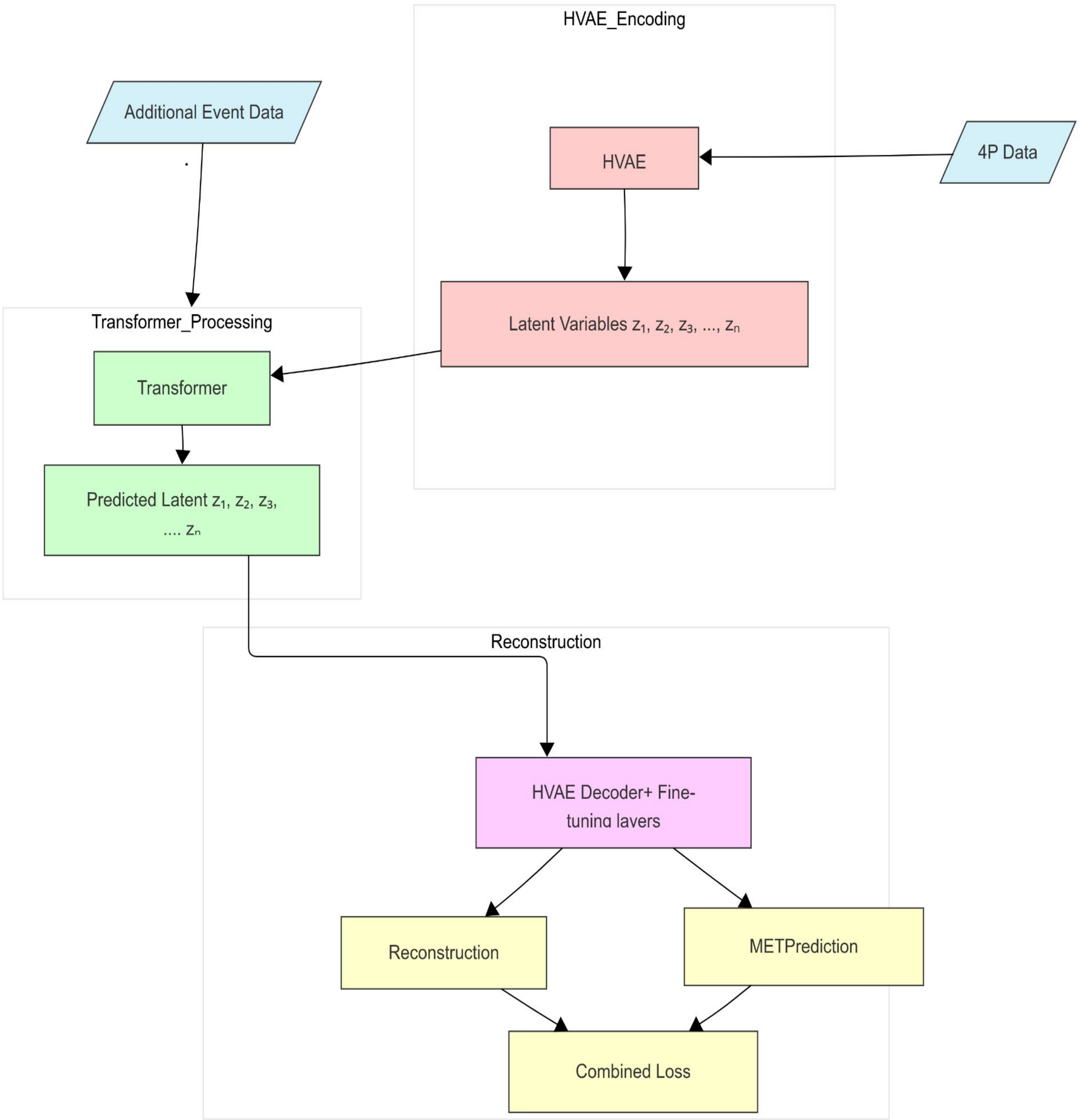
This proposal suggests adding a new model into the BEAD's pre-existing codebase.This model is designed by substituting d**iscrete latent models (e.g., VQ-VAE)** in the research paper (link) with a **Hierarchical Variational Autoencoder (HVAE)** for a **semi-continuous latent space**, complemented with a **Transformer** for **anomaly detection from HEP jet data**. The **HVAE pre-trains** on **background events** to learn fundamental physics, while the **Transformer** employs **partial latent data** and context to find more profound patterns. **Multi-task fine-tuning** then transfers the model to **detect anomalies** through **reconstruction** and **missing energy prediction**.

The method utilizes **unlabeled data**, decreasing simulation dependence and domain shift handling. It constructs a **base mode**l that captures **shallow** and **deep physics**, increasing sensitivity to new physics signals. In contrast to previous work centered on **classification**, this addresses **anomaly detection** directly, providing a **new workflow** for **HEP research**.

3.1 Workflow

- **Pre-train HVAE** on **background jet data** with **VAE loss (reconstruction + KL)**.
- Freeze **HVAE encoder**, pass **partial latent** to **Transformer** with event context.
- Train **Transformer** to **predict** remaining **latent** using **distance loss**.
- Fine-tune **HVAE decoder** with **Transformer output** for **reconstruction** and **MET** prediction.
- Score events for **anomalies** based on **reconstruction error** and **MET** deviation.

3.2 Model Architecture



4.Software Contribution

I will instantiate the **HVAE-Transformer** model in **PyTorch** to keep it modular by introducing a new class for the HVAE-Transformer model in **BEAD's models.py** file, which way is combining with BEAD'S already **existing codebase**. This enables **smooth reuse** and **expansion** within the BEAD framework.The **input data** needed for training this model is **identical** to other already **existing models** in **BEAD**.

The **difference** will be in the **loss function**, I'll add a **new loss function** for **MET Loss** and **Combined Loss**. Also Add **code** to **train model** and **detect anomalies** on the **Combined Loss** function.**Weights & Biases (WandB)** could also be added for **automated experiment tracking**, enhancing BEAD's research capabilities.**Dockerization** will ensure **portability** across systems, **packaging** BEAD's dependencies.

Example Code snippet for HVAETransformer model

```
class HVAETransformer(nn.Module):
    def __init__(self, input_dim, hidden_dim, d_model, n_heads):
        super(HVAETransformer, self).__init__()
        self.hvae_encoder = nn.Sequential(nn.Linear(input_dim,
            hidden_dim), nn.ReLU())
        self.transformer = nn.TransformerEncoderLayer(d_model=d_model,
            nhead=n_heads)
        self.decoder = nn.Linear(hidden_dim, input_dim)
        self.met_head = nn.Linear(hidden_dim, 1)  # MET prediction

    def forward(self, x, context):
        latent = self.hvae_encoder(x)
        trans_out = self.transformer(torch.cat([latent, context], dim
            =-1))
        recon = self.decoder(trans_out)
        met = self.met_head(trans_out)
        return recon, met, latent
```

Example Code snippet for Loss function

```
def met_loss(pred_met, true_met):
    return nn.MSELoss()(pred_met, true_met)

def combined_loss(recon, x, pred_met, true_met, latent, kl_weight=1.0):
    recon_loss = nn.MSELoss()(recon, x)
    met_loss_val = met_loss(pred_met, true_met)
    kl_div = -0.5 * torch.mean(1 + latent - latent.pow(2) - latent.exp
        ())
    return recon_loss + met_loss_val + kl_weight * kl_div
```

Example Code snippet for training

```
model = HVAETransformer(input_dim=4, hidden_dim=64, d_model=128,
    n_heads=4)
optimizer = torch.optim.Adam(model.parameters(), lr=1e-3)

for epoch in range(10):
    recon, met, latent = model(data, context)  # data, context from
        BEAD's loader
    loss = combined_loss(recon, data, met, true_met, latent)
    optimizer.zero_grad()
    loss.backward()
    optimizer.step()

    # Anomaly detection
    anomaly_score = loss.item()  # High loss indicates anomaly
    print(f"Epoch {epoch}, Loss: {loss.item()}, Anomaly Score:
        {anomaly_score}")
```

5.Tentative Timeline

Week 1 - 3	Phase 1	• Setup environment, explore HEP data, and design HVAE architecture. • Pre-train HVAE on background data and validate reconstruction.
Week 4 - 5	Phase 2	• Implement Transformer, integrate with HVAE latent, and train prediction task.
Week 6	Midterm Evaluation	• Submit midterm deliverables and gather mentor feedback. • Complete any remaining blog posts and documentation
Week 8 - 9	Phase 3	• Add fine-tuning layers, train multi-task objectives, and test anomaly scoring. • Tune model for high accuracy.
Week 10 -11	Phase 4	• Evaluate the model . • Compare with other existing models and draw conclusions.
Week 12	Final Submission	• Complete final blog posts and document all aspects of the project
Post GSoC		• Continue working at the organization and contributions in HEP research.

6.Conclusion

6.1 Progress Tracking

I will have a **daily task logger** of **tasks, challenges**, and **insights** in a shared document. **Weekly progress meetings** with mentors will **track milestones** and make plans accordingly. **Daily Slack communication** will provide **instant feedback** on **doubts** and **outcomes**, keeping the project on track.

6.2 Future scope

As an open-source effort, I'd pursue a **research paper** if results show strong anomaly detection performance, **sharing findings** with the **HEP community.** I'm also interested in becoming a **maintainer**, supporting future enhancements like **larger-scale pre-training** or additional physics tasks.

6.3 Outlook

- Proposes a novel **HVAE-Transformer** model for **HEP anomaly detection.**
- Enhances the project with **modular**, **reproducible** software tools.
- Aims to bridge **physics** and **ML**, opening doors to **new discoveries**.

7.Programming background

My **coding experience** started in **12th grade** with **C++**, which provided a solid foundation in **logic** and **structure**. In college, I switched to **Python** due to its **flexibility**, learning it through projects such as **"Stock Transformer"** with **PyTorch**, which further enhanced my knowledge of **deep learning**. I've since applied **Python** and **PyTorch** to diverse **AIML** projects, including **VAEs** for **drug innovation** and **reinforcement learning** for **traffic systems**, sharpening my skills in **model implementation** and **optimization**; key for contributing effectively to BEAD's anomaly detection framework.

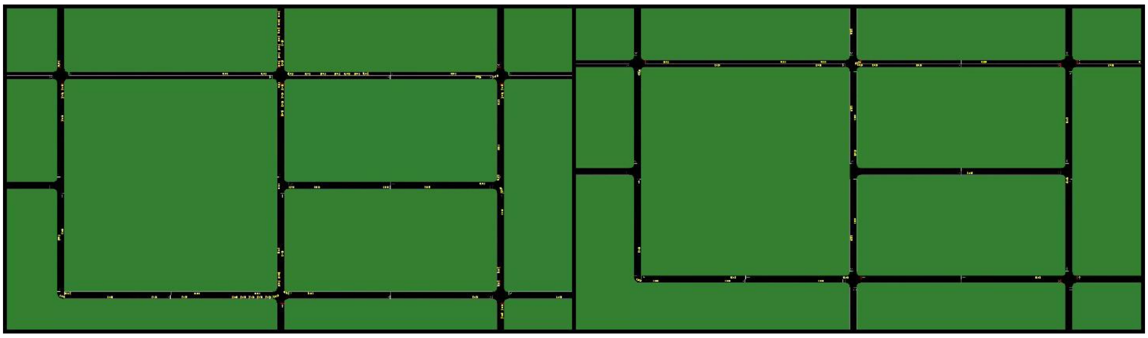
7.1 Projects

7.1.1 Stock Transformer

- I developed a deep learning model for **stock price prediction** using **vanilla Transformer** for prediction and **Time2Vec** for data pre processing .
- I used **Yfinance** module for **financial data**. Later I tested the model with different stocks , below are the results for **TATAMOTORS**
- Project Blog :https://abhi-shekkk.github.io/Stock_Transformer/blog
- **Test MSE:** 871.0430
- **Test RMSE:** 29.5134
- **Test MAPE:** 3.73%

7.1.2. Smart Flow: AI based Intelligent Traffic Control System

- Aim of the project was to make a **AI powered traffic light system** that optimizes signal timings based on real time vehicle congestion.
- I've tested many **computer vision techniques** (Canny Edge Detection, Semantic Segmentation, Hough Transform) to separate lanes for real time **vehicle count detection**.
- Trained SUMO on a **Reinforcement Learning model** that dynamically adjusts traffic signals timings, in this I've achieved an **32% reduction in wait time** and reduced emission by an average of **27%**.



8.References

GSoc Participation

I have **not** participated in **GSoC** before as such but I'm quite eager to explore and contribute to the world of open source.

Project BEAD under **CERN-HSF** is my **primary** and **sole focus** for **GSoC** this year.

References

- https://arxiv.org/abs/2401.13537v3
- https://arxiv.org/abs/2007.03898
- Improving Variational Autoencoders for New Physics Detection at the LHC with Normalizing Flows
- https://arxiv.org/abs/2105.14027
- https://arxiv.org/abs/2312.14190
- https://medium.com/towards-data-science/hands-on-anomaly-detection-with-variational-autoencoders-d4044672acd5
- https://arxiv.org/pdf/1706.03762
- [2202.03772] Particle Transformer for Jet Tagging