VILNIAUS UNIVERSITETAS
MATEMATIKOS IR INFORMATIKOS FAKULTETAS
INFORMATIKOS INSTITUTAS
KOMPIUTERINIO IR DUOMENŲ MODELIAVIMO KATEDRA

Kompiuterinio modeliavimo II kurso magistro darbas

Aukštos energijos fizikos duomenų generavimas naudojant giliojo mokymosi metodus

High energy physics data generation using deep learning methods

Atliko:
Paulius Balčiūnas parašas

Vadovas:
Asist. dr. Tomas Raila

Vilnius
2024

Turinys

Santrauka **3**

Summary **4**

Įvadas **5**

1. Susijusių darbų apžvalga **8**

2. Mašininis mokymas **11**
2.1. Mašininio mokymo modeliai 12
2.1.1. Variacinis auto-enkoderis (VAE) 12
2.1.2. Generatyvūs priešiškieji tinklai (GAN) 15
2.1.3. Normalizuojantys srautai 17
2.1.4. Gauso maišos modelis 18
2.2. Duomenų vertinimo metrikos 20
2.2.1. *Frechet inception* atstumas (*FID*) 20
2.2.2. Frobenijaus norma 20
2.2.3. Fizikinės metrikos 21
2.3. Aktyvacijos funkcijos 22
2.4. Optimizavimo algoritmai 24
2.5. Paketo normalizavimas 25

3. Duomenų rinkinys **26**
3.1. Duomenų rinkinio savybės 27

4. Duomenų generavimo eksperimentai **29**
4.1. *VAE* eksperimentai 29
4.2. *GAN* duomenų generavimo eksperimentai 37
4.3. Normalizavimo srautų duomenų generavimas 42
4.3.1. Modelio treniravimas ir naujų duomenų generavimas 43
4.4. Rezultatų ir greitaveikos palyginimas 48
4.5. *HEP* duomenų generavimas pagal Gauso maišos modelį (*GMM*) 51
4.5.1. *Gauso* maišos modelio duomenų generavimo rezultatai 52
4.5.2. *GMM* palyginimas su gilaus mokymo modelio rezultatais 54

Išvados **55**

Ateities tyrimų planas **57**

Literatūros šaltiniai **58**

Įvykę susitikimai su darbo vadovu **61**

Santrauka

Magistro darbe yra siekiama sukurti fizikinių dalelių įvykių duomenų generavimo sprendimą, galintį padėti *CERN* organizacijai atlikti aukštos energijos fizikinių dalelių tyrimus. *CERN* organizacijos mokslinėje veikloje yra naudojami *Monte Carlo* metodu paremti fizikinių dalelių duomenų generatoriai. *Monte Carlo* generatoriai veikia neefektyviai, nes naudoja dideliais kiekiais kompiuterinę atmintį ir daug procesoriaus laiko atliekant naujų duomenų generavimo skaičiavimus. Magistro darbe bus analizuojami tradicinių mašininio mokymo ir gilaus mokymo modeliai. Atlikus analizę, bus pritaikytas modelis, kuris galėtų, naudodamas mažiau kompiuterinės atminties bei procesoriaus laiko, sugeneruoti aukštos energijos fizikinių dalelių įvykių duomenis pagal *Monte Carlo* metodu sugeneruotą fizikinių dalelių duomenų rinkinį. Modelis duomenų generavimo procese, pateiks naujus fizikinius dalelių duomenis, kurių savybės yra panašios į pasirinktą duomenų rinkinį, kuris yra sugeneruotas *Monte Carlo* metodu. Modelio duomenų generavimo kokybė bus įvertinta pagal fizikinės invariantinės masės, dalelių greičio, *Frechet Inception* atstumo ir Frobenijaus normos metrikas. Mašininio mokymo modelio veikimas yra ištirtas pagal jo veikimo laiką ir kompiuterinių resursų naudojimą. Darbe yra paskaičiuoti dydžiai, žymintys kiek laiko mašininio mokymo modelis užtruko ir kiek panaudojo kompiuterinės atminties atliekant naujų duomenų generavimo užduotį.

Summary

Darbo pavadinimas kita kalba

The aim for the master's thesis project is to develop a particle physics data generation solution that could assist *CERN* organization science researchers in performing high energy particle physics research work. *CERN* uses *Monte Carlo* method based data generators for high energy particle data generation. The generators use too much computer memory and processor time for data generation computations. In the master degree project classical machine learning and deep learning methods will be researched. After the performed analysis a machine learning model will be applied that uses less computer memory and processor time for particle physics data generation. The generated data will be calculated without any critical statistical flaws and based on the statistical qualities that are found in the *Monte Carlo* method generated dataset. The used machine learning method data generation results will be analyzed with invariant mass calculation, transverse momentum, *Frechet Inception* and *Frobenius* norm metrics. As well, the studied machine learning model computation time and memory usage will be calculated and studied to see if the applied machine learning model uses computer resources effectively.

Įvadas

Duomenų modeliavimo metodai yra vieni iš svarbiausių priemonių, siekiant suprasti tam tikrų pasaulio reiškinių veikimo prasmę. Metodai yra naudojami meteorologijoje, fizikoje, ekonomikoje, chemijoje ir t.t. Meteorologijos srities specialistai taiko sudėtingas diferencialines lygtis bandydami išsiaiškinti oro klimato sąlygų susidarymo kilmę. Fizikai arba astronomai siekdami suvokti galaktikų ir kosminių kūnų atsiradimo priežastis, apibrėžia atsiradimo procesus tam tikrais dėsniais pagal fizikines taisykles [6]. Duomenų modeliavimas aktyviai yra naudojamas mašininio mokymo srityje klasifikuojant bei generuojant tam tikrus naujus duomenis, siekiant išsiaiškinti jų prasmę, kilmę ir panaudojimą. Mašininio mokymo industrijoje yra aktyviai naudojami diskriminatyvūs (*angl. k. diskriminative*) ir generatyvūs (*anglų k. generative*) duomenų modeliavimo mašininio mokymo metodai. Diskriminatyviajame duomenų modeliavime yra siekiama taikyti metodus, galinčius tikimybiniais, matematiniais skaičiavimais sugeneruoti duomenų etiketes ir pagal jas klasifikuoti kokio nors duomenų rinkinio įrašus į grupes. Tai būtų tiesinis klasifikatorius, sprendimų medžiai arba logistinė regresija. Generatyviajame duomenų modeliavime, taikomi duomenų modeliai, tam, kad ištirti kaip pasaulyje gali tam tikri duomenys atsirasti [6]. Vieni iš žinomiausių metodų būtų generatyvūs priešiškieji tinklai (*anglų k. Generative Adversarial Networks, trumpiau GAN)*, variaciniai auto enkoderiai (*anglų k. Variational Auto Encoders, VAE*), normalizavimo srautai (*anglų k. Normalizing Flows*) ir pan. Mašininio mokymo duomenų modeliavimo srityje, siekiant tiksliai ir efektyviai sumodeliuoti duomenis, reikia efektyviai naudoti kompiuterio aparatinę įrangą, kurioje yra naudojami modeliai. Mašininio mokymo modeliais modeliuojant duomenis, turėtų būti taikomi kompiuterio procesoriai (*CPU*) ir grafinės kortos (*GPU*) [10], [25]. Grafinės kortos skaičiuoja greičiau negu kompiuterio procesorius. Naudojant kompiuterinę grafinę kortą, galima tikėtis geresnio skaičiavimo greičio ir tikslumo, nes tai suteikia galimybę lygiagrečiai atlikti tam tikrus mašininio mokymo modelių skaičiavimus. Kompiuterio procesoriai naudingi mašininio mokymo modelių procesams, tačiau juose atliekami duomenų modeliavimo skaičiavimai būna kartais lėti, nes naudoja mažiau branduolių ir yra labiau tinkami sinchroniniams procesams vykdyti. Naudojant mašininio mokymo duomenų modeliavimo sprendimus, turėtų būti atsižvelgiama į jų veikimą, efektyvumą ir naudą. Branduolinės fizikos tyrimų organizacijos *CERN* kompiuterinių skaičiavimų laboratorijose yra taikomi*Monte Carlo* metodu paremti duomenų generatoriai, skirti generuoti naujų aukštos energijos fizikinių dalelių (*HEP*) įvykių duomenis, kurie prognozuoja, kokius *CERN* mokslininkai fizikinius dalelių duomenis galėtų užfiksuoti didžiajame *CERN* organizacijos greitintuve (*anglų k. Large Hadron Collider, LHC*). Taikomi duomenų generatoriai yra paremti *Monte Carlo* metodo veikimu ir generuoja *ATLAS* ir *CMS* detektorių aukštos energijos dalelių įvykių duomenis. *CERN* dalelių įvykių duomenys yra paremti fizikiniais modeliais, kurie galėtų būti keturių leptonų, viršutinių kvarkų formavimo ir pan. Sugeneravus naujus fizikinius duomenis, *CERN* mokslininkai gali tikslingai apsiskaičiuoti fizikines prognozes, kurios informuotų, kokių rezultatų gali mokslininkai tikėtis po tam tikro sėkmingo *LHC* greitintuvo paleidimo, nes *LHC* greitintuvo paruošimas moksliniams eksperimentams kainuoja daug finansinių išteklių ir aparatinės įrangos paruošimo laiko.

CERN organizacija naudoja naujų fizikinių duomenų generavimo *Monte Carlo* generatorius pasaulinėje *LHC* greitintuvo tinklo (*anglų k. Worldwide LHC computing grid*, trumpiau *WLCG*) kompiuterinėje platformoje. Sugeneruoti nauji duomenys yra naudojami palyginimui su teoriniais fizikiniais duomenimis, kurie apskaičiuoti pačių platformos naudotojų. Sugeneruoti nauji duomenys padeda tiksliau apskaičiuoti naujus, tam tikrų dalelių [1] fizikinius matavimus. *CERN* generatorių *ATLAS* ir *CMS* duomenų skaičiavimo generatoriai per metus bendrai sugeneruoja 10^{10}

kiekio duomenų *WLCG* paslaugos naudotojams. Sugeneruotų dalelių duomenų kiekis yra apytiksliai tris kartus didesnis negu tikrų, *LHC* greitintuvo užfiksuotų duomenų [1]. Norint, kad tiek fizikinių duomenų *Monte Carlo* generatoriai sugeneruotų, išnaudojama daug *WLCG* skaičiavimo tinklo procesorių atminties ir veikimo laiko. Padidėjus *WLCG* platformos *Monte Carlo* skaičiavimų algoritmų kompiuterinių resursų naudojimo apkrovai, platformos vartotojai galėdavo gauti riboto, kiekio *Monte Carlo* būdu sukurtų naujų fizikinių aukštos energijos dalelių duomenų. Jeigu sistemos vartotojai gaudavo riboto kiekio fizikinius dalelių duomenis, vartotojams atlikus fizikinius skaičiavimus, galėdavo pastebėti juose netikslumų, kurie atsiranda tam tikrose apskaičiuotuose dalelių fizikiniuose įverčiuose. Apskaičiuoti įverčiai yra susiję su tam tikrų dalelių *Z Higgs bozono* dydžių skaičiavimais, iš branduolio pabėgusių dalelių greičių tyrimais ir pan. Todėl kartais *CERN* organizacijoje sutrikdavo fizikinių dalelių tyrimai. Kadangi *CERN* organizacija turi didelių pasiekimų dalelių nuotraukų generavimo srityje taikant mašininio mokymo metodus [24], mokslininkai pradėjo paieška alternatyvių sprendimų, kurie galėtų nenaudodami daug kompiuterių procesoriaus laiko ir atminties, sugeneruoti naujus ir tikslius fizikinių dalelių duomenis.

Magistro darbe siekiama taikyti naujų fizikinių duomenų generavimo sprendimą naudojant giliojo mokymo modelius. Sprendimas bus parengtas pagal modelių naujų duomenų generavimo eksperimentus taikant pasirinktam *Z Higgs* aukštos energijos dalelių įvykių duomenų rinkiniui. Atlikus modelių eksperimentus, modelių rezultatai bus tarpusavyje palyginami. Gilaus mokymo rezultatai dar bus palyginami su atliktais mašininio mokymo rezultatais gautais mokslo tiriamajame darbe. Giliojo mašininio mokymo metodai, kurie bus taikomi yra *Variational Auto encoder*, *Beta-Variational Autoencoder*, *Generative Advesarial Networks*, *Normalizing Flows*. Mokslo tiriamajame darbe mašininio mokymo metodai, kurie buvo taikomi yra *gauso* miksurų (anglų k. *Gaussian Mixtures*) modelio. Vykdant eksperimentus bus analizuojamas modelių veikimas ir duomenų generavimo rezultatų sukūrimas. Rezultatai bus vertinami pagal tam tikras statistines ir fizikines metrikas kaip *KL* divergencija, *Wasserstein* atstumas, invariantinė masė ir pan. Atlikus eksperimentus, bus siekiama atrasti mašininio mokymo sprendimą, kuris galėtų būti taikomas efektyvaus *CERN HEP* duomenų generatoriaus sukūrimui. Naujas duomenų generavimo sprendimas naudos mažiau kompiuterinės atminties, procesoriaus laiko bei apskaičiuos tiksliai naujus aukštos energijos, fizikinius dalelių įvykių duomenis.

Tikslas

Pritaikyti gilaus mokymo modelį, galintį sugeneruoti naujus fizikinius duomenis, kurių savybės yra panašios į pasirinkto duomenų rinkinio, vertinant pagal tam tikras statistines ir fizikines metrikas

Uždaviniai

1. Atlikti *Monte Carlo* metodu sugeneruoto duomenų rinkinio savybių analizę.
2. Išanalizuoti giliojo mokymo modelių veikimą, skirtų generuoti naujus duomenis.
3. Atlikti duomenų generavimo eksperimentus, taikant pasirinktus giliojo mokymo modelius.
4. Išanalizuoti gautus giliojo mokymo modelių duomenų generavimo eksperimentų rezultatus pagal pasirinktas duomenų vertinimo metrikas.
5. Palyginti mokslo tiriamojo darbo projekto rezultatus, gautus pagal paprastus mašininio mokymo metodus, su naujais duomenų generavimo rezultatais, kurie yra gauti taikant giliojo mokymo metodus.

Naudota mokslo tiriamojo darbo teorinė ir praktinė informacija

- Skyrelyje 3 pateikta *Z Higgs* 4 leptonų duomenų rinkinio savybės, kurios buvo aprašytos mokslo tiriamajame darbe.
- Skyrelyje 2.1.4 pateikta gauso maišos teorija, kuri buvo aprašyta mokslo tiriamajame darbe.
- Skyrelyje 4.5 pateikta atliktų gauso maišos modelio eksperimentų rezultatai, kurie buvo aprašyti mokslo tiriamajame darbe.
- Skyrelyje 2 įvadinė mašininio mokymo teorinė sąvokų informaciją.

1. Susijusių darbų apžvalga

Efektyvus mašininio mokymo duomenų generavimo sprendimas, turi nenaudoti daug kompiuterinės įrangos procesoriaus veikimo greičio, atminties bei privalo, pagal tam tikras metrikas, tikslingai skaičiuoti aukštos energijos dalelių įvykių duomenis generavimo metu. Modelis, turi sugeneruoti tinkamas duomenų generavimo tikimybines aproksimacijas, pagal kurias būtų kuriami tikslūs, nauji aukštos energijos dalelių duomenys. Norint sužinoti, kaip būtų galima pasiekti šį tikslą, šiame skyriuje yra pateikiami išanalizuoti tam tikrų mokslinių duomenų generavimo darbų pavyzdžiai.

Straipsnyje *Particle Cloud Generation with Message Passing Generative Adversarial Networks* autoriai aptaria gilaus mokymo *MPGAN* modelio veikimą [18]. *MPGAN* yra naujų duomenų generavimo modelis, kuris yra paremtas *GAN* veikimo principais. Taikomas generuoti naujus fizikinių dalelių susidūrimų ir skilimo įvykius, kuriuos pagal, autorių aprašytą, aukštos energijos fizikinių dalelių duomenų rinkinį, *JetNet*. Įvykių informaciją sudaro dalelių dušo (*anglų k. particle shower*) procesas, kuris yra susietas su dalelių branduolių skilimu ir hadronizacija, kuri formuoja tam tikras fizikines daleles. Procesų dalelių duomenys yra šviesaus kvarko, viršutinio kvarko ir gluano. Duomenų rinkinyje yra dalelių koordinatės, greitis ir dvejetainė reikšmė, kuri yra reikalinga klasifikuoti dalelių duomenis. Autoriai straipsnyje mini, kad *Monte Carlo* generatoriai negali sugeneruoti šių duomenų, nes yra sudėtingi, turintys daug sąryšių ir vaizduojantys nemažai procesų, kuriuos sudaro daug duomenų. Straipsnyje autoriai pabrėžia, kad nėra mašininio mokymo modelio, kuris buvo pritaikytas būtent generuoti naujus *JetNet* duomenis. Autoriai sukūrė savo *GAN* tipo modelio architektūrą *MPGAN*. Straipsnyje aprašoma, kad modelis buvo apmokytas pagal mokymo aibę ir rezultatai įvertinti taikant *Frechet Inception*, *Wasserstein*, minimalaus panašumo ir dengimo metrikas. Straipsnio autoriai dar pateikia sugeneruotų dalelių duomenų histogramas, kuriose parodo, kiek tam tikro duoto rinkinio stulpelių reikšmių skirstinys yra panašus į sugeneruotų duomenų stulpelio. Apskaičiavus metrikų rezultatus, autoriai palygina su pasirinktais mašininio mokymo modelių *r-GAN*, *TreeGAN* ir *GraphCNN* rezultatais. Straipsnio išvadose yra minima, kad *MPGAN* gali duoto duomenų rinkinio skirstinį sugeneruoti pagal sugeneruotų duomenų aproksimacijas, tik negali iš jų sukurti naujus fizikinius dalelių susidūrimų ir skilimų duomenis. Autoriai testavo kompiuteriniame įrenginyje, kuriame yra prieinamas procesorius ir grafikos korta *NVIDIA A100 GPU*. Įvertino modelio greitaveiką ir pastebėjo, kad generuojant kiekvieną fizikinių dalelių įvykį užtrunka 35,7μs laiko. Kitame kompiuteriniame įrenginyje, testavo *Monte Carlo* generatorių, kuris gali veikti tik su kompiuterio procesoriumi. Įrenginyje buvo pasiekiamas procesorius, turintis 8 branduolius. *Monte Carlo* generatorius vieną dalelių įvykį sugeneruoja per 46 ms. Tai parodo, kad autorių atrastas mašininio mokymo modelis *MPGAN* pasiekia perspektyvių rezultatų, kuriuos galėtų autoriai tobulinti ateityje.

Straipsnio *Machine Learning methods for simulating particle response in the Zero Degree Calorimeter at the ALICE experiment, CERN* [12] autoriai, pristato eksperimentus, kuriuose siekė sugeneruoti simuliacinius *CERN* organizacijos *Zero degree Calorimeter* eksperimento duomenis taikant mašininio mokymo metodus. Pagrindinė eksperimentų idėja yra atrasti efektyvų mašininio mokymo modelį, kuris galėtų sugeneruoti fizikinio eksperimento simuliacinius duomenis duomenų analizės darbams. Duomenis vaizduoja detektoriuje užfiksuotų dalelių įvykių energijas. *CERN* kompiuterinių skaičiavimų laboratorijose eksperimento duomenys yra generuojami pagal *GEANT4 Monte Carlo* metodo grįstą programinę įrangą, kuri neefektyviai naudoja kompiuterinius *CERN GRID* infrastruktūros resursus. Straipsnio autoriai siekė atrasti naują efektyvų duomenų generavimo sprendimą. Duomenų generavimo tyrimams pasirinko *VAE* ir *DC-GAN* modelius.

Straipsnio įvade autoriai pabrėžia mintį, jog *Monte Carlo* eksperimentai reikalauja didelių kompiuterinių resursų kiekių skaičiavimams *CERN GRID* kompiuterinėje infrastruktūroje, sudarančią 500000 kompiuterių procesorių 170 mazguose. Pradinė duomenų aibė, kuri yra naudojama *CERN Monte Carlo* duomenų eksperimentams turi 8 milijonus daleles ir jų energijas apibūdinančių įrašų. Autoriai pirmiausia taikė eksperimentuose mašininio mokymo klasifikatorių, išfiltruoti duomenis pagal tam tikrą kriterijų, kurie galėtų būti naudingi duomenų generavimui. Vėliau, duomenų generavimo eksperimentu metu, pasirinko mašininio mokymo modelius, kurie gali sugeneruoti naujus duomenis [12]. Vėliau sugeneruoti duomenys buvo atvaizduoti 44 x 44 pikselių iliustracijoje, kurioje turėjo parodyti tikslingai apskaičiuotą išskirtą dalelių energiją detektoriaus pluoštuose. Atlikus eksperimentus, autoriams pavyko pasiekti pakankamai gerus rezultatus. Sugeneravo duomenis, kuriuose galima pastebėti dalelių energijos bruožus, paremtais tam tikrais fizikiniais dėsniais. Nauji sugeneruoti duomenys buvo išanalizuoti *Frechet Inception* ir *Wasserstein* atstumu metrikomis. Autoriai rezultatuose pastebėjo, kad *VAE* galėjo sukurti detektoriaus užfiksuotu energijų pozicijas, bet iliustracijos gavosi neryškios. *GAN* modelis galėjo sugeneruoti geresnes iliustracijas, tik jose nėra tinkamai užfiksuoti dalelių duomenų sąryšiai. Panaudojus pridėtinį regresoriaus metodą (*anglų k. auxillary regressor*), pavyko pagerinti sugeneruotų duomenų rezultatus taikant *DC-GAN* modelį.

Taikant, *B-VAE* modelį, galima gauti duomenis, kurie gali būti tikslesni negu *VAE* modelio sugeneruoti duomenys. Straipsnyje "Event generation and statistical sampling for physics with deep generative models and a density information buffer" [23], autoriai aprašo mašininio mokymo duomenų generavimo eksperimentus, kuriuos atliko taikant standartinį *GAN*, *DijetGAN*, *VAE* ir *B-VAE* modelius. Eksperimentuose buvo siekiama nustatyti ar modeliai galėtų sėkmingai generuoti naujus aukštos energijos fizikinių dalelių įvykių duomenis pagal *Monte Carlo* metodu sugeneruotus 4 leptono ir viršutinių kvarkų duomenų rinkinius. Straipsnyje autoriai mini, kad atlikus duomenų generavimo eksperimentus, pavyko nustatyti, jog standartiniai *VAE* ir *GAN* modeliai yra mažiau naudingi duomenų generavime negu *B-VAE* arba *DijetGAN* modeliai. *VAE* modeliai suformuoja latentinėje erdvėje duomenų, tikimybines aproksimacijas, kurios gali sugeneruoti aukštos energijos dalelių duomenis, neturinčius standartinio normalaus skirstinio. Tačiau, jeigu *VAE* modelis latentinės erdvės duomenų aproksimacijų skaičiavimuose pridėtų papildomai sugeneruotą atsitiktinę, tikimybinę seką, turinčią Gauso skirstinio savybes, tuomet *VAE* modelio sugeneruoti rezultatai pasidaro geresni. Sugeneruotų duomenų rezultatų, savybės pasidaro panašesnės į *Monte Carlo* metodu sugeneruotų duomenų savybes. Autoriai lygino pasirinktų *Monte Carlo* metodu duomenų energijų, dalelių greičių ir koordinačių skirstinius pagal histogramas. Nustatė, kad skirstiniai yra beveik panašūs, pagal rezultatų grafikus, galima pastebėti, kad duomenų skirstinių histogramos beveik vienodos. *VAE* modelio variantas, kurio funkcionalumas yra papildytas atsitiktiniu triukšmo generuojančią funkciją, pavadintas *B-VAE* modeliu. Autoriai nustatė, kad *B-VAE* modelis tinkamai sugeneruoja aukštos energijos dalelių įvykių koordinates p_x, p_y, p_z. Standartiniams *GAN* modeliams yra sudėtinga sumodeliuoti dalelių įvykių azimuto kampą ϕ. *B-VAE* modelis gali sugeneruoti aukštos energijos dalelių įvykių duomenis, pagal kuriuos galima sėkmingai galima apskaičiuoti dalelių greičius p_T, poliarinius θ ir azimuto ϕ kampus.

Mašininio mokymo srities tyrėjai straipsnyje "DijetGAN: A Generative-Adversarial Network Approach for the Simulation of QCD Dijet Events at the LHC" pasiūlė duomenų generavimo sprendimą, kuris gali pateikti tikslius, efektyvius, sintetinius aukštos energijos dalelių duomenis [20]. Moksliniams *LHC* tyrimams, reikalingi dideli palyginimams skirti fizikiniai duomenų rinkiniai. *CERN* organizacija aktyviai ieško simuliacinių duomenų sprendimų, kurie galėtų efektyviai sukurti protonų (*pp*) susidūrimų įvykių duomenis. Straipsnio autoriai ištyrė, kad šiai proble-

mai spręsti gali padėti gilaus mokymo duomenų modelis *GAN (Generative Adversarial Network)*, kuris pradžioje buvo skirtas generuoti nuotraukas, vėliau buvo po truputį taikomas simuliacinių duomenų generavimui. Atliekant duomenų generavimo eksperimentus, straipsnio autoriai mini, kad naudojo transformuotą protonų susidūrimų rinkinį, kuriame tam tikrais skaičiavimais gaunasi dalelių įvykių azimuto kampas lygus 0. Pritaikius šį apdorojimo sprendimą, duomenys tampa labiau tinkami naujų duomenų generavimui naudojant *GAN* modelį. Eksperimentų metu, *GAN* modelis transformuoja tam tikrą sudarytą seką atsitiktinių skaičių į duomenis, kurie turėtų fizikinę prasmę apibūdinančią protonų įvykių duomenis. Vėliau, šie duomenys buvo naudojami *GAN* modelio diskriminatoriaus metode, kuris įvertina ar sugeneruoti duomenys yra panašūs į duotą *Monte Carlo* metodu sugeneruotą protonų susidūrimų duomenų rinkinį, kuris gautas pagal metodo tankumo funkciją *pdf*. Autoriams pritaikius *GAN* modelio paremtą dalelių įvykių generavimo sprendimą, pavyksta sugeneruoti tinkamus naujus dalelių įvykių duomenis. Atliktuose eksperimentuose, autoriai nustatė, kad sugeneruotų dalelių įvykių duomenų histogramos beveik sutampa ir *GAN* modelio diskriminatoriaus ir generatoriaus netekties reikšmės sukonverguoja į vieną tikimybę $0,7$, kuri rodo, kad modelis galės gerus duomenis sugeneruoti. Straipsnyje yra minimas eksperimentas, kuriame autoriai patikrino, jog *GAN* modelis gali sugeneruoti 1 milijoną dalelių susidūrimų duomenų, kurie gali būti pagal invariantinės masės metriką laisvai lyginami su *Monte Carlo* būdu sugeneruotu rinkiniu. Taikant *GAN* modelį, autoriams pavyko sugeneruoti 1 milijoną fizikinių dalelių įvykių per 80 sekundžių grafikos kortoje *NVIDIA Quadro P6000*.

Norint pasiekti geresnį duomenų generavimo tikslumą, nebūtina turėti pilnai apmokytus mašininio mokymo metodus. Straipsnyje "Event generation with normalizing flows" [4], autoriai pateikia giliojo mokymo *normalizing flows* modelį, kuris galėtų būti tinkamas *CERN* aukštos energijos dalelių duomenų generavimui. Modelis gali sugeneruoti duomenis, turinčius *Drell Yann LHC* procesų tipą. Autoriai pastebėjo, kad pastiprintų sprendimo medžių ir *GAN* modeliai sugeneruoja naujus duomenis, panaudojant daugiau kompiuterinių resursų naujų duomenų operacijoms. Tai keldavo problemų asinchroniniams aukštos energijos dalelių duomenų skaičiavimams, kurie siekdavo inferencijos metu apskaičiuoti aproksimacijai skirtą *Jacobian* funkcijos reikšmę. *NICE* metodas skaičiuodamas naujus aukštos energijos dalelių duomenis, netaiko gradientų skaičiavimų. Autoriams atlikus eksperimentus naudojant *NICE* modelį, pavyko gauti rezultatus, kurie yra geresni, negu tam tikrų paprastų duomenų skaičiavimo algoritmų. Įvykdžius duomenų generavimo procesą naudojant *NICE* metodą, autoriai palygino gautus rezultatus su kitais duomenų generavimo rezultatais gautais iš *SHERPA* generatoriaus, kuriame duomenų generavimo skaičiavimai yra atliekami naudojant *Monte Carlo* metodą. Autoriai nustatė, kad mašininio mokymo *Normalizing Flow* metodas sugeneruoja aukštos energijos dalelių duomenis, kurie yra tinkamo tikslumo ir turi savybių, kuriomis pasižymi *Monte Carlo* metodu sugeneruoti duomenys. Duomenų generavimo eksperimentuose, pagal *NICE* metodą, straipsnio [4] autoriai nustatė, kad *NICE* metodas yra pajėgus sugeneruoti naujus fizikinius duomenis, kurie gali įgauti *LO* ir *NLO* fizikinių skaičiavimų tipų tikslumą.

Atlikus panašių darbų analizę, pastebėta, kad taikomo naujų duomenų generavimo modelio rezultatus, gali nulemti duomenų rinkinys, pagal kurį yra apmokamas modelis, jo architektūros struktūra ir metrikų įverčiai. Magistriniame darbe pasirinkta generuoti naujus duomenis taikant *GAN*, *VAE* ir normalizuojančių srautų modelius. Modelių rezultatai bus įvertinti pagal *Wasserstein* ir *Frechet Inception* atstumo metrikas. Patikrinami rezultatų duomenų skirstinių histogramos ir įvertinta greitaveika. Atlikus modelių rezultatų analizę, bus nustatyta, kuris iš pasirinktų modelių, galės būti taikomas naujų aukštos energijos fizikinių duomenų generavimui.

2. Mašininis mokymas

Mašininis mokymas (*angl. machine learning*) - automatiniai, kompiuterizuoti metodai, kurie leidžia sistemoms automatiškai sukurti pačioms uždavinių sprendimus be kokių nors reikalingų, specifinių užprogramuotų instrukcijų. Loginius uždavinius sprendžia mašininio mokymo metodai struktūriniu ir nestruktūriniu duomenų modeliavimu būdu. Struktūriniai duomenys būtų lentelės, o nestruktūriniai duomenys būtų duomenys, kurie nėra sudėlioti tam tikra griežta apibrėžta tvarka [21].

Mašininio mokymo modelių yra paprastų, kuriuos sudaro paprasti metodai, atliekantys paprastesnes analitines duomenų prognozių užduotis. Metodai nenaudoja sudėtingų architektūrų ir yra lengvai apmokomi pagal tam tikrus treniravimo duomenis ir hiperparametrus. Pavyzdžiui, tiesinę regresiją, logistinę regresiją, Gauso maišos modelis ir pan.

Tačiau, yra mašininio mokymo modelių, kurie yra sudėtingesni ir yra kaip poaibis bendros mašininio mokymo srities, paremti sudėtingomis architektūromis. Modeliai naudoja neuroninius tinklus, kurie atlieka tam tikras duomenų transformacijas, reikalingas kokiems nors loginiams uždaviniams spręsti. Gilaus mokymo modeliai naudojami nuotraukų ir kalbos atpažinimui, teksto sintezei, trūkstamų vietų nuotraukose užpildyti, objektų atpažinimui ir pan. Mašininio mokymo metodai yra skirstomas į šias kategorijas:

- Prižiūrimasis mokymas (*angl. supervised learning*) - mašininio mokymo sritis, kurioje metodai atlieka automatinį mokymosi procesą pagal duotą mokymo duomenų rinkinį. Rinkinyje yra vektorių poros, kurios nariai apibūdina duomenų rinkinyje naudojamą įvestį (*angl. Input Vector, feature, points*) ir norimo gauti rezultato duomenis(*angl. expected value*). Srityje yra tikslas pagal įvesties duomenis nustatyti, kokia funkcija buvo naudojama norint gauti rezultato duomenis pagal įvesties ir rezultato duomenų vektorių poras apskaičiuojant prognozes. Šios mašininio mokymo srities kategorijai priklauso modeliai: tiesinė regresija, polinominė regresija, multilinijinė regresija, logistinė regresija ir pan.

- Neprižiūrimasis mokymas (*angl. unsupervised learning*) - mašininio mokymo sritis, kurioje modeliai automatiškai panaudoja paruoštus pradinius duomenis ir nenaudoja papildomų duomenų, kurie informuoja modelius, kokie rezultatai turi būti gaunami. Šios srities modeliai, pagal duotus pradinius duomenis, siekia automatiškai nustatyti jų struktūrą. Modeliai taikomi klasifikavimui ir naujų duomenų generavimui. Neprižiūrimo mokymo sričiai priklausytų mašininio mokymo modeliai: *Gausian Mixtures*, *k-means*, *Kernal Density Estimation (KDE)* ir pan.

- Sustiprintasis mokymas (*angl. reinforcement learning*) - mašininio mokymo sritis, kurioje modeliai naudoja įvesties duomenų rinkinius spręsti tam tikrus loginius uždavinius kokioje nors aplinkoje arba situacijoje. Spręsdami loginius uždavinius, sustiprintojo mokymo modeliai patobulėja ir gali dar geriau išspręsti kitus loginius uždavinius.

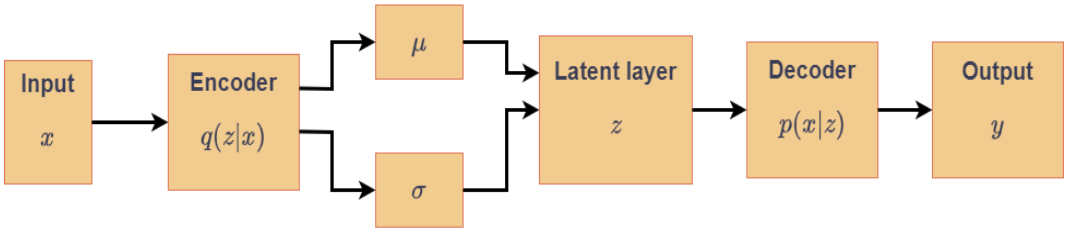
2.1. Mašininio mokymo modeliai

Darbe bus naudojami gilaus mokymo variaciniai auto-enkoderiai (*VAE*), B-variaciniai auto enkoderiai (*B-VAE*), generatyvūs priešingi tinklai(*GAN*) ir normalizuojančio srauto (*Normalizing flow*) modeliai. Dar ir eksperimentuose bus taikomas tradicinis, mašininio mokymo Gauso maišos modelis (*Gaussian Mixtures*). Šioje darbo aprašo dalyje yra aprašomi naudotų mašininio mokymo modelių teoriniai veikimo principai. Skyriuje apžvelgiama, pagal kokius metodus pasirinkti modeliai apskaičiuoja naujus duomenis Y, pagal duotą duomenų rinkinį X.

2.1.1. Variacinis auto-enkoderis (VAE)

Gilaus mokymo modelis, kuris panaudojęs tam tikrą duotą duomenų rinkinį X, pagal neuroninį tinklą, transformuoja duomenis į pavidalą, kuris priklausytų latentinei erdvei Z ir vėliau iš jos galėtų sugeneruoti naujus duomenis Y, taikant neuroninių tinklų metodus.

Variacinis auto-enkoderis yra tinkamas nuotraukų triukšmo šalinimui ir trūkstamų vietų nuotraukose užpildymui.*VAE* modelis gali transformuoti pradinius duomenis ir generuoti naujus duomenis naudojant metodus: enkoderis $q(z|x)$ ir dekoderis $p(x|z)$. Enkoderis yra skirtas apskaičiuoti naujų duomenų aproksimacijas vidurkį μ ir kovariacijos matricą σ, kurie yra panaudojami surasti latentinės erdvės z reikšmes, turinčias Gauso skirstinio savybes. Modelis, apskaičiavęs z reikšmes, perduoda į dekorio metodą, kuris pagal neuroninius tinklus, sugeneruoja naujus duomenis (1 pav.).



1 pav. Naujų duomenų generavimo procesas taikant VAE modelį

Norint surasti z latentinės erdvės reikšmes, turi būti atliktas reparemetrizacijos etapas. *VAE* modelio reparametrizacijos etape, naudojant vidurkį μ ir kovariacijos matricą σ, surandamos z erdvės reikšmės taikant duomenų aibę ϵ, apskaičiuota pagal funkciją, grąžinančią "atsitiktinio triukšmo" reikšmes. ϵ reikšmės yra normalaus Gauso skirstinio. z skaičiuojama pagal formulę (2.1 formulė), kuri yra viena iš tinkamiausių atliekant *VAE* latentinės erdvės reparametrizacijos skaičiavimus [7]:

$$z = \mu + \sigma\epsilon \tag{2.1}$$

VAE apskaičiavęs μ ir σ, dar taiko netekties funkcijos skaičiavimui $\mathcal{L}(\theta, \phi, x^{(i)})$ [7]. θ yra *VAE* išmokstamų parametrų aibę, kurią sudaro svorių matricos w ir papildomi nariai b. ϕ žymi papildomų narių aibę, kurie žymimi b. *VAE* netekties funkcija yra apibrėžiama formule:

$$\mathcal{L}(\theta, \phi, x^{(i)}) = -D_{KL}(q_\phi(z|x^{(i)}|p(z)) + \frac{1}{n}\sum_{j=0,i=0}^{n,n}(x^{(i)} - y^{(i)})^2 \tag{2.2}$$

$(x^{(i)} - y^{(i)})^2$ apskaičiuoja atkūrimo netekties reikšmę, vadinama vidutine kvadratine paklaida (*anglų k. Mean Square Error*). Skaičiuojama pagal įvesties ir išvesties rinkinių x ir y skirtumų

kvadratų sumą padalintą iš duoto duomenų rinkinio įrašų ilgio n. $-D_{KL}(q(z|x^{(i)}|p(z))$ apskaičiuoja *KL* divergencijos atstumą pagal modelio apskaičiuotas aproksimacijas μ ir σ. *KL* divergencijos atstumo formulė 2.3:

$$D_{KL}(q(z|x^{(i)}|p(z)) = \frac{1}{2}\sum_{j=1}^{J}(1 + \log((\sigma_j^{(i)})^2) - (\mu_j^{(i)})^2 - exp((\sigma_j^{(i)})^2)) \tag{2.3}$$

Treniruojant *VAE* modelį yra siekiama, kad *KL* divergencijos ir vidutinės kvadratinės paklaidos reikšmės mažėtų, tam kad modeliui pavyktų geriau sugeneruoti naujus duomenis, turinčius panašių savybių į duotų duomenų. Naujuose duomenyse savybės turi pasižymėti duotų duomenų netiesiškumu, koreliacija ir skirstiniais. Duomenų rinkinių panašumas tarp duoto ir naujo yra vertinamas naudojant W_1 (*Wasserstein*) atstumą, *FID* panašumą, frobenijaus normą ir pan. Modelis, taikydamas netekties funkciją $\mathcal{L}(\theta, \phi, x^{(i)})$, suranda gradientus, modelio mokymo ciklo metu taikant stochastinį gradientų mažėjimo algoritmą (*Adam* arba *Stochastinis gradiento mažėjimas*). Apskaičiuoja momentinius vektorius m ir v, kurie yra kaip koeficientai, reikalingi mažinti apskaičiuotas *VAE* modelio reikšmes μ, σ ir y, tam kad būtų arti duoto duomenų rinkinio reikšmių,

σ ir μ vertės yra apskaičiuojamos naudojant *VAE* modelio enkoderio metodą. σ ir μ yra surandami pagal linijinės transformacijos formulę [19]:

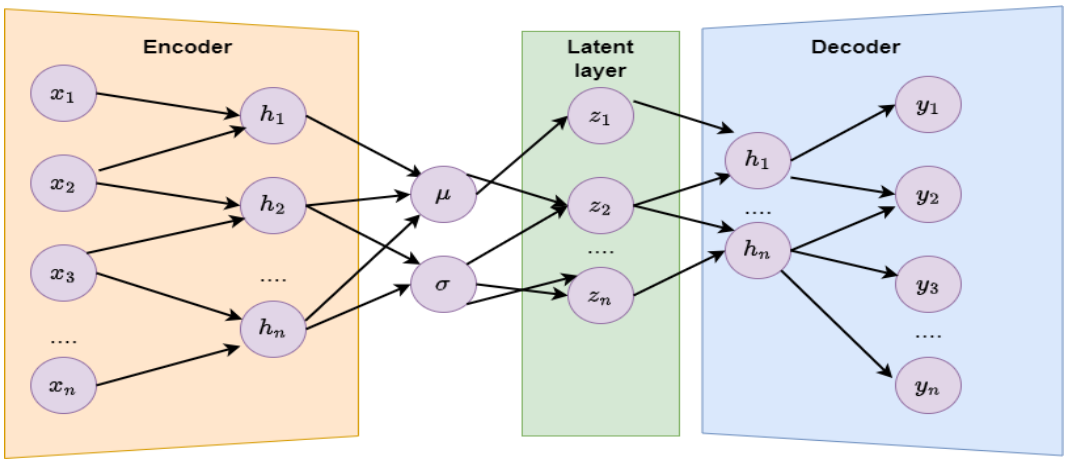
$$y = xw^T + b \tag{2.4}$$

A^T yra svorių matrica ir papildomas narys b. w yra tiesinės transformacijos svorių matrica, kuri gali būti inicializuojama atsitiktiniais skaičiais ir iš jos apskaičiuojama transponuojama matrica w^T. b yra papildomas narys (*bias unit*), kurio reikšmė irgi gali būti inicializuojama atsitiktiniu skaičiumi. b koeficientas yra reikalingas padaryti, kad bent vienas neuronas būtų aktyvus neuroninio tinklo sluoksnyje ir galėtų vis dar gautis netiesiškumas gilaus mokymo tinkle, jeigu svorių sandauga tarp x ir A^T sukuria matrica, kurioje reikšmės mažos, artėjančios link nulio, su maža dispersija. Matrica A dimensija gali būti bet kokia ir yra apibrėžiama vartotojo. A matricos eilučių ilgis yra toks pat, kaip duoto duomenų rinkinio. Pilnos *VAE* modelio μ ir σ formulės:

$$\mu = xw_1^T + b_1, \sigma = xw_2^T + b_1 \tag{2.5}$$

Modelis, pritaikęs linijinės transformacijos metodą, pritaiko dar aktyvacijos funkciją tam kad skaičiuojami duomenys turėtų netiesiškumo bruožų. Aktyvacijos funkcijos gali būti *ReLU*, *ELU*, *LeakyRelu* ir pan. Plačiau apie aktyvacijos funkcijas, galima paskaityti skyrelyje 2.3.

Apskaičiavus μ ir σ, *VAE* modelis suranda latentinės erdvės z reikšmes, taikant reparametrizacijos metodą 2.1. Vėliau latentinė erdvės reikšmės z yra perduodamos į dekoderio metodą $p(x|z)$, kuris irgi naudoja tiesinės transformacijas ir aktyvacijos funkcijas naujų duomenų y generavimui iš latentinės erdvės reikšmių z (2 pav.).



2 pav. Naujų duomenų generavimo procesas taikant *VAE* modelio enkoderio ir dekoderio metodus. x_n žymi duoto rinkinio vektorius. h_n reikšmės, kurios yra gautos pagal tiesinės transformacijos metodą ir aktyvacijos funkciją. y_n naujų duomenų vektoriai,

2.1.2. Generatyvūs priešiškieji tinklai (GAN)

Mašininio mokymo modelis, kurį sudaro multi-sluoksnio *perceptrono* modeliai - generatorius G ir diskriminatorius D. Generatoriaus modelis G, sukuria naują duomenų skirstinį pagal duotą duomenų rinkinį X. Diskriminatoriaus modelis D, bando nustatyti, skaičiuodamas tikimybę, ar sugeneruoti nauji duomenys yra tokie pat kaip duoto duomenų rinkinio duomenys X. *GAN* modelio treniravimo metu, siekiama maksimizuoti tikimybę, kurią apskaičiuoja diskriminatoriaus modelis. Kuo labiau aukštesnė apskaičiuota diskriminatoriaus modelio tikimybė, tuo sugeneruoti nauji duomenys yra labiau panašūs į duoto duomenų rinkinio duomenis. Toks modelio veikimas vadinamas $min - max$ žaidimu. Unikalus sprendimas G ir D modeliams egzistuoja. D duomenų analizavimo tikimybė yra maždaug beveik lygi $\frac{1}{2}$. Apibrėžia, kad diskriminatoirus $D(x)$ nebegali atskirti ar sugeneruoti duomenys yra iš generatoriaus $G(x)$, ar yra duoti duomenys. Tuomet, gaunasi geresni, nauji sugeneruoti duomenys pagal $G(x)$ metodą.

Pats priešingų tinklų modelis yra treniruojamas grįžtamojo perdavimo metodu (*anglų k. backpropogation*. Priešiškų tinklų modelio veikimas tarsi primintų "padirbtų pinigų nusikaltimą". Generatoriaus modelis visame *GAN* modelio veikimo metu siekia sugeneruoti naujus duomenis, kurie tarsi būtų "padirbti pinigai". Diskriminatoriaus modelis, būtų "policija", kuri bando nustatyti ar rasti pinigai yra padirbti ar tikri. Norint tinkamai apmokyti pačius priešingų tinklų modelius, reikia taikyti grįžtamojo perdavimo (*backpropogation*) ir *Dropout* metodus.

Modelio treniravimo pradžioje yra sugeneruojama duomenų aproksimacijų skirstinys $p(z)$ pagal generatoriaus modelį G ir atsitiktinę sugeneruotą aibę z, kurios skirstinys primintų normalaus skirstinio triukšmą [11]. z aibės duomenys yra susiejami su generavyvaus modelio parametrais θ, apibrėžiant $G(z, \theta)$. θ, tai parametrai, kuriuos generatoriaus modelis, apskaičiuoja treniravimo cikle pagal linijinės transformacijos metodus ir aktyvacijos funkcijas $p(z)$ duomenų transformavimui. Vėliau sugeneruoti duomenys yra paduodami diskriminatoriaus modeliui $D(x, \theta)$, kuris paskaičiuoja tikimybę, rodančią ar sugeneruoti duomenys, pagal jų savybes, galėtų būti tokie pat kaip duoto duomenų rinkinio X. Bendras generatoriaus G ir diskriminatoriaus D apmokymas veiktų kelių elementų netekties funkcijoje $V(D, G)$, kuri apibrėžta forma:

$$V(D, G) = \mathbb{E}_{x \sim p_{data}(x)} [\log D(x)] + \mathbb{E}_{z \sim p_z(z)} [1 - \log D(G(z))] \tag{2.6}$$

Netekties funkcijos $D(x)$ ir $G(x)$ gali būti skaičiuojamos pagal binarinės kryžminės entropijos metodą (*anglų k. Binary cross entropy*) naudojant sugeneruotus triukšmo z duomenis ir duotą duomenų rinkinį x. Norint, kad modelio netekties funkcija $G(z, \theta)$ sugeneruotu tikslius duomenis, formulės netekties funkcijos vertė $\log D(G(z)$ turi gautis aukšta. Kuo aukštesnė netekties vertė, tuo labiau modelis D teigia, kad $G(x)$ modelio sugeneruoti duomenys yra X. $D(x)$ patikrinęs sugeneruotus duomenis $p(z)$, atnaujinami modeliams $G(x)$ ir $D(x)$ skirti gradientai, naudojant treniravimo optimizavimo algoritmą ir vykdoma kita modelių treniravimo epochą, kuriame vėl $G(x)$ generatorius sugeneruoja naują triukšmą $p(z)$ kitai treniravimo epochai. Jeigu modeliui $G(x)$ pavyksta sugeneruoti duomenis naudojant triukšmo aibę z, tuomet jo netekties reikšmė mažėja. Jeigu $D(x)$ modeliui pavyksta neatskirti $G(x)$ sugeneruotų duomenų nuo duotų duomenų, tuomet diskriminatoriaus $\log D(G(z)$ reikšmė didėja. Priešingu atveju mažėja, kai nustato, jog $G(x)$ reikšmės gavosi $p(z)$ reikšmės, pašios į duotų duomenų rinkinį X.

Funkcijos reikšmė $V(D, G)$ turi gautis neigiamą $-V(D, G)$ pagal generatoriaus $G(x)$ ir $D(x)$ netekčių funkcijų reikšmes. Tai galės pasakyti, kad modeliams $G(x)$ ir $D(x)$ pavyko sugeneruoti naujus duomenys, kurie yra panašūs į duotą duomenų rinkinį.

Generatyviųjų priešiškųjų tinklų treniravimo procesas pagal stochastinio gradiento algoritmą:

---

for epochų kiekis **do**

for k žingsniai **do**

Sugeneruojama triukšmo m aibė $z = z^{(1)}, ..., z^{(m)}$, kuri turi triukšmo skirstinį $p_g(z)$

Parenkamas vektoriai aibei p_{data} iš duomenų rinkinio X duomenų generavimui

Atnaujinama diskriminatoriaus latentiniai kintamieji pagal gradientą:

$$\nabla_{\theta_d} \frac{1}{m} \sum_{i=1}^{m} \left[\log D(x^{(i)}) + \log(1 - D(G(z^{(i)}))) \right] \tag{2.7}$$

end for

Sugeneruojama triukšmo m aibė $z = z^{(1)}, ..., z^{(m)}$, kuri turi triukšmo skirstinį $p_g(z)$

Atnaujinami generatoriaus latentiniai kintamieji pagal gradientą

$$\nabla_{\theta_d} \frac{1}{m} \sum_{i=1}^{m} \left(1 - D(G(z^{(i)})) \right) \tag{2.8}$$

end for

Vienas iš siekių treniravimo metu, padaryti, jog skaičiuojamos gradiento struktūros "konverguotų". Būtų pasiektas reikiamas apskaičiuotų modelio generatoriaus ir diskriminatoriaus parametrų minimumas, nuo kurios nebesikeistų latentinių kintamųjų rezultatai, reikalingi skaičiuoti duotų duomenų aproksimacijas tiek generatoriuje G, tiek diskriminatoriaus modelyje D. Rezultatas, kuris turi būti pasiektas modelio treniravimo metu, $p_g = p_{data}$. Tai reikštų, kad sugeneruotų duomenų modelio G skirstinys p_g yra lygus duomenų rinkinio X skirstiniui p_{data}.

Norint, tokį rezultatą pasiekti reikia daug eksperimentų atlikti, parinkti tinkamas modelių parametrus linijinių transformacijų dalyse bei pritaikyti tinkamus hiperparametrus, kurie galėtų padėti pasiekti lygybės rezultatą $p_g = p_{data}$. Straipsnyje "Generative Adversarial Nets" [11], kurį parašė vienas iš jo autorių *Ian J. Goodfellow* yra minima, kad *GAN* modeliui ne visada pavyksta lygybės rezultatą pasiekti ir šiomis dienomis yra aktyviai ieškomą sprendimų kaip tokį lygybės rezultatą pasiekti.

2.1.3. Normalizuojantys srautai

Neparametrinis mašininio mokymo modelis, kuris yra taikomas naujų duomenų generavimui. Modelį sudaro srauto sluoksniai (*anglų k. flow layers*), naudojami rasti latentinius kintamuosius z, taikant duotą duomenų rinkinį X ir neuroninių tinklų linijines transformacijas. Modelis suradęs latentinius kintamuosius, sugeneruoja naujus duomenis Y, kurių savybės panašios į duotą duomenų rinkinį X. Normalizuojančių srautų modelis yra naudojamas generuoti naujas nuotraukas pagal kokias nors duotas nuotraukas. Modelis gali užpildyti tam tikras trūkstamas vietas nuotraukose ir generuoti naujus garso takelius pagal duotus garso duomenis. Modelio treniravimo metu, apskaičiuojama tikimybinė modelio netekties funkcija. Netekties funkcijos skaičiavimo metu, mažinamos aproksimuotos latentinių kintamųjų reikšmės, kurios yra naudojamos generuoti naujus duomenis Y. Mažėjant latentinėms reikšmėms z, modeliui gali pavykti geriau sugeneruoti naujus duomenis Y [2].

Normalizuojančio srauto modelio jungiamasis sluoksnis yra duomenų transformavimo metodas, kuris duotą duomenų rinkinį X padalija į du lygius poaibius X_A ir X_B ir atlieka duomenų transformavimo skaičiavimus X_B poaibiui. X_A lieka nepakeistas, tik normalizuojančio srauto modelis, priskiria latentinės erdvės kintamiesiems z ir keičia X_B poaibį transformuojant pagal neoroninių tinklų linijines transformacijas, aktyvacijos funkcijas ir poaibį X_B [**?**]. Poaibiai priklauso intervale $(x_A, x_B) \in R^d \cdot R^{D-d}$

$$f(x) = \begin{cases} z_A = x_A \\ z_B = h(z_A, \theta(x_B)) \end{cases} \tag{2.9}$$

$h(z_A, \theta)$ žymėtų jungiamąją funkcija, kurioje yra naudojama neuroninių tinklų linijinės transformacijos funkciją θ. Jungiamasis srautas atvaizduoja reikšmes principu $h(z, \theta) : R \to R$. Funkcija $h(x, \theta)$ suprojektuoja reikšmes, kurios yra duotų duomenų aproksimacijos, priklausančios realiųjų skaičių aibei.

Normalizuojančio srauto modelio mokymo cikle, gali būti taikomas netekties funkcijos metodas, *KL* divergencijos, vidutinės kvadratinės paklaidos ir pan (2.3 formulė). Tik eksperimentuojant su skirtingomis netejties funkcijomis, reikia atkreipti sugeneruotų duomenų rezultatus pagal tam tikras pasirinktas statistines metrikas, kurios įvertina duomenų kokybę. Ištirti, ar sugeneruotuose duomenyse yra geros savybės *Wasserstein* atstumo, frobenijaus normos, *FID* reikšmės ir pan. Plačiau apie metrikas 2.2 skyrelyje.

Idėja netekties funkcijos tokia pat, kaip *VAE* ir *GAN* modelio. Modelio treniravimo metu turi būti irgi mažėjanti netekties funkcijos vertė naudojant parametrų optimizavimo algoritmą, kuriame apskaičiuojamos netekties funkcijos išvestinės, iš kurių yra surandamos gradiento reikšmės. Gradientai yra reikalingi surasti tikimybę $p(z)$, pagal kurią sugeneruoja naujus duomenis Y. Mažėjant netekties funkcijos vertei, gali didėti tikimybės $p(z)$ reikšmės, jeigu tinkami optimizavimo algoritmo parametrai yra parinkti, kurie sėkmingai mažina netekties funkcijos vertę. Jeigu pasibaigus modelio mokymo aibei, nėra apskaičiuota tinkama funkcijos reikšmė, didiname modelio treniravimo epochų skaičių. Plačiau apie tai nagrinėjame praktinėje darbo dalyje.

Norint pagerinti normalizavimo srautų sugeneruotų duomenų rezultatus Y, galima taikyti afininės transformacijos srautus (vadinami auto regresyvūs modeliai). Padidina apskaičiuotas normalizavimo srautų tikimybes, transformuojant latentinės erdvės z reikšmes, padidina naujų duomenų generavimo tikslumą. Darbe yra naudojama maskuotas auto regsyvus modelis, kuris duomenų transformacijos metu, naudoja bitų matricą, kurią padauginus su transformuotais duomenimis z, atsiranda daugiau atsitiktinių, nevienodų reikšmių naujuose duomenyse [8].

2.1.4. Gauso maišos modelis

Gauso maišos modelis (*anglų k. Gaussian Mixtures*, *GMM*) yra neprižiūrimojo mašininio mokymo parametrinis modelis [13], kuris pagal duotą duomenų rinkinį X, suranda tikimybinius Gauso skirstinius, kurie apibūdina tam tikras duomenų rinkinio X dalis (vadinami klasteriais) [27], [3]. Modelis yra tinkamas klasifikavimo ir naujų duomenų generavimui pagal tam tikrą duotą duomenų rinkinį X.

1. **Klasifikavimo užduoties sprendimas pagal *gauso* maišos modelį**
   Gauso maišos modelis apskaičiuoja tikimybinius svorius $p(x|k)$ klasifikavimo funkcijos $\hat{f}(x)$ ieškojimo procese. Svoriai yra gaunami iš, *EM* (*Expectation Maximization*) algoritmo būdu, nustatytos x_i aibės vidurkių μ, π_k įverčių bei x_i kovariacijos Σ matricos [9].

   (a) *Gauso* maišos modelio veikimo pirmame žingsnyje inicializuoja Σ matricą, kurios vertės lygios 1, π_k tikimybės, kurių pradinės vertės lygios $\frac{1}{k}$ (k - svorių kiekis) bei *M* kiekis centroidžių, kurios yra parenkamos pagal *k-means*, *k-means++* algoritmą arba atsitiktiniu būdu iš duomenų rinkinio matricos X.

   (b) Modelio veikimo antrame žingsnyje, pagal bendrinę tikimybinių svorių formulę, apskaičiuojami $p(k|x_i)$ x_i taškų klasterių narystės svoriai (2.10 formulė):

$$p(k|x_i) = \frac{\pi_k N(x|\mu_k, \Sigma_k)}{\sum_{k=1}^{K} \pi_k N(x|\mu_k, \Sigma_k)} \tag{2.10}$$

$$N(x_i|\mu_k, \sigma_k^2) = \frac{1}{\sqrt{2\pi * \sigma_k^2}} \exp\left(-\frac{(x_i - \mu_k)}{2\sigma_k^2}\right) \tag{2.11}$$

   (c) Trečiame modelio veikimo žingsnyje, *EM* algoritmo būdu perskaičiuojamos Σ, μ ir π_k vertės taikant formules 2.12, 2.13, 2.14 bei antrame žingsnyje surastus pirminius tikimybinius svorius $p(k|x_i)$:

$$\mu = \frac{\sum_{i=1}^{n} P(k|x_i) x_i}{\sum_{i=1}^{n} P(k|x_i)} \tag{2.12}$$

$$\sigma^2 = \frac{\sum_{i=1}^{n} P(k|x_i)(x_i - \mu_k)^2}{\sum_{i=1}^{n} P(k|x_i)}; \Sigma = (\sigma_1^2, \sigma_2^2, .., \sigma_K^2) \tag{2.13}$$

$$\pi_k = \frac{\sum_{i=1}^{n} P(k|x_i)}{n} \tag{2.14}$$

   Svarbu dar skaičiuojant tikimybinius $p(k|x_i)$ svorius duomenų rinkinio vektorių x_i klasterio narystės nustatymui, apskaičiuoti pirmines tikimybes π_k tenkinančias sąlygas:

$$0 \leq \pi_k \leq 1, \;\; \sum_{k=1}^{K} \pi_k = 1$$

(d) Ketvirtame žingsnyje, kartoti antrą ir trečią modelio veikimo žingsnius iki tol, kol $p(k|x_i)$ svoriai nebekis.

EM algoritmas yra uždaros formos ir iteratyviai taikomas modelio klasterių sudarymo procese. Apkroksimacijos įverčių ieškojimo procedūroje, modelis cikliniu būdu *k* kartų surandą vidurkio μ ir Σ matricos reikšmes. Dar įverčiai yra ieškomi naudojant tikimybių skaičiavimo logaritmo *Maximum likelihood* metodu (2.15 formulė).

$$L = \log P(X|\pi, \mu_k, \Sigma_k) = \sum_{k}^{n} log \sum_{i}^{K} N(x_i|\mu, \Sigma) \tag{2.15}$$

Metodas *Maximum likelihood* metodas logaritminiu būdu padaro, kad $p(k|x_i)$ svorių tikimybės būtų didesnės ir nebūtų mažos. Tada bus tikslesni klasifikavimo užduoties rezultatai.

Apskaičiavus *p(x)* įverčius, Gauso maišos modelis, gali nustatyti, kokiems priklauso klasteriams tam tikri duomenų rinkinio x_i įvesties vektoriai. Klasterių kiekis apibrėžtas konstantą *k*, o *Python scitkit-learn* bibliotekos komandose apibrėžiamas kintamuoju *n_components*. Modelis tinkamas duomenų rinkinio anomalijų aptikimui, vaizdo ir nuotraukų kompresijai, duomenų klasifikavimui ir naujų duomenų generavimui.

2. **Duomenų generavimas pagal *Gauso* maišos modelį**
Norint išbandyti modelį su naujais, nematytais duomenimis, galima taikyti modelio duomenų generavimo algoritmą. *Gauso* maišos modelis, gali suteikti galimybę sugeneruoti deterministinių, atsitiktinių skaičių aibę y_i, kuri įgautų daugiamačio, *Gauso* skirstinio savybes. Generuojant skirstinį, Gauso maišos modelis apskaičiuoja μ, π_k ir σ_k įverčius, pagal kuriuos transformuoja sugeneruotą, Gauso savybių įgavusią, atsitiktinių skaičių turinčią aibę y_i. μ, π_k ir σ_k yra apskaičiuojami pagal duotą duomenų rinkinį X ir 2.12, 2.13, 2.14 formules.

2.2. Duomenų vertinimo metrikos

Norint įvertinti modelio veikimo kokybę, naudojamos rezultatų vertinimo metrikos. Darbe bus naudojama *Frechet* atstumo metrika įvertinti savybių panašumą tarp originalaus duomenų rinkinio ir sugeneruotų duomenų rinkinio. *Wasserstein* atstumas bus naudojamas nustatyti kokio reiktų svorio kiekio, leidžiančio sugeneruotiems duomenims būti panašiems į duotą duomenų rinkinį. Papildomai darbe bus naudojamos fizikinės metrikos, pagal kurias bus norima įvertinti ar sugeneruotų, naujų duomenų dalelių fizikinės savybės yra tinkamos. Fizikinės metrikos yra fizikinė invariantinė masė ir dalelių pabėgimo greitis.

2.2.1. *Frechet inception* atstumas (*FID*)

Analitinė metrika, skirta įvertinti panašumą tarp tam tikrų aibių vektorinių taškų. Metrikos veikimo tikslas nustatyti, kiek duotas vektorių taškų rinkinys P yra panašus į duotą rinkinį Q. Literatūroje metrika yra dar vadinama, kaip *Wasserstein-2* metrika. *FID* darbe yra naudojama rezultatų duotų fizikinių duomenų palyginimui su naujais sugeneruotai duomenims. *Fréechet inception* atstumo formulė yra apibrėžiama tokia formule:

$$d((\mu_P, \sigma_P), (\mu_Q, \sigma_Q)) = ||\mu_P - \mu_P||_2^2 + Tr(\sigma_P + \sigma_Q - 2(\sigma_P * \sigma_Q)^{\frac{1}{2}}) \tag{2.16}$$

μ_P, μ_Q - duotų duomenų rinkinio P ir išvesties vektoriaus Q apskaičiuoti vidurkiai. σ_P, σ_Q yra apskaičiuotos duoto duomenų rinkinio P ir išvesties vektoriaus Q kovariacijos matricos. $Tr(x)$ yra linijinės algebros, matricos funkcija apibrėžta formule [14]:

$$Tr(x) = \sum_{i=0}^{n-1} x_{11} + x_{22} + ... + x_{n-1n-1} \tag{2.17}$$

Funkcija apskaičiuoja pagrindinę, matricos diagonalės įverčių sumą.

2.2.2. Frobenijaus norma

Analitinė, matricų duomenų vertinimo metrika, taikoma analizuoti, skaitine verte, dviejų duotų matricos reikšmių panašumą. Frobenijaus metrika yra tinkama analizuoti duotų tam tikrų duomenų rinkinių P ir Q savybių panašumą pagal koreliacijų matricas. Koreliacijų matricos gali būti apskaičiuotos naudojant Pearsono arba Spearmeno metodus. Frobenijaus metrika yra suformuluojama formule [15]:

$$||M||_F = \sqrt{\sum_{i,j} |a_{ij} - b_{ij}|^2} \tag{2.18}$$

Frobenijaus metrika žymima $||M||_F$. a ir b yra dviejų skirtingų koreliacijų matricų P ir Q elementai. Analizuojant statistinius rezultatus, svarbu nustatyti, jog apskaičiuota frobenijaus reikšmė būtu artėjanti link 0.

$$\lim_{x \to 0} M(x) \tag{2.19}$$

Kuo mažesnė frobenijaus metrikos reikšmė, tuo labiau analizuojamos matricos yra panašios.

2.2.3. Fizikinės metrikos

Metrikos, kurios bus naudojamos įvertinti sugeneruotų, naujo duomenų rinkinio fizikinių duomenų kokybė būtų invariantinė masė m_{inv} ir jėgos impulsas p_T [26], [23].

- **Invariantinė masė**

  m_{inv} invariantinė masė, tai yra dalelės masė, kuri išlieka po aukštos energijos dalelių susidūrimo ir branduolio skylimo. Aukštos energijos dalelės invariantinė matrica skaičiuojama formule:

$$m_{inv} = \sqrt{(\sum_{n=1}^{4} E_i)^2 - (\sum_{n=1}^{4} p_{x_i})^2 - (\sum_{n=1}^{4} p_{y_i})^2 - (\sum_{n=1}^{4} p_{z_i})^2} \tag{2.20}$$

  E žymi aukštos energijos dalelių energija, matuojama *eV*(Elektron voltai). p_x, p_y ir p_z dalelių įvykių koordinatės, žyminčios susidūrimų ir branduolio skylimo. Fizikiniai vienetai m_{inv} yra *GeV* giga-elektro voltai.

- **Fizikinių dalelių greitis**

  Dalelių p_T greitis, pamatuoja, kokiu greičiu atvyksta iki susidūrimo įvykio ir pabėga iš tam tikros dalelės branduolio. Formulė yra apibrėžiama keliais variantais:

  Naudojant p_i dalelės koordinates (x, y).

$$p_T = \sqrt{p_x^2 + p_y^2} \tag{2.21}$$

  p_x, p_y, p_z yra tam tikros dalelės (x, y, z) koordinatės. p_T vienetai yra *GeV/s* giga elektrovoltai per sekundę.

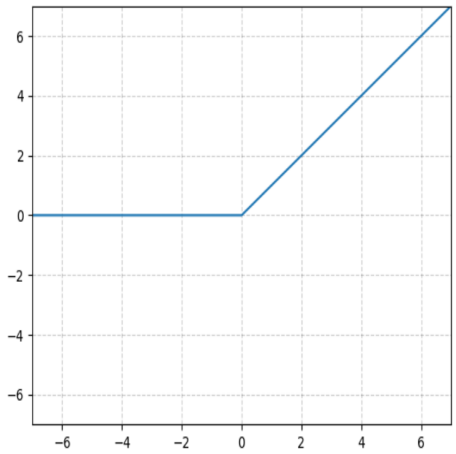
2.3. Aktyvacijos funkcijos

Gilaus mokymo modelių apmokymo procesuose yra naudojamos aktyvacijos funkcijos, skirtos panaikinti dingstančiojo gradiento skaičiavimo problemą. Tai būtų situacija, kai susiformuoja mažos linijinių transformacijų svorių matricos reikšmės dėl gradientų reikšmių. Susidaro labai mažos gradientų reikšmės, artėjančios link 0 [21]. Tokiu momentu, taikant gilaus mokymo modelį generuoti naujus duomenis, gali nesusidaryti netiesiškumo požymiai naujuose sugeneruotuose duomenyse. Norint, kad pavyktų gilaus mokymo modeliui tiksliau sugeneruoti naujus duomenis, taikomos modelio architektūroje aktyvacijos funkcijos:

- *ReLU* aktyvacijos funkcija:

$$ReLU(x) = max(0, x) \tag{2.22}$$

  Funkcija įvestyje naudoja reikšmę x, kuri yra bet kokio tipo skaitinė reikšmė ir taikoma $max()$ funkcijoje. Jeigu x reikšmė yra didesnė už 0, tada $ReLU(x)$ grąžina x reikšmę [21], [17], 3 pav. Kitu atveju $ReLU$ grąžintų 0.

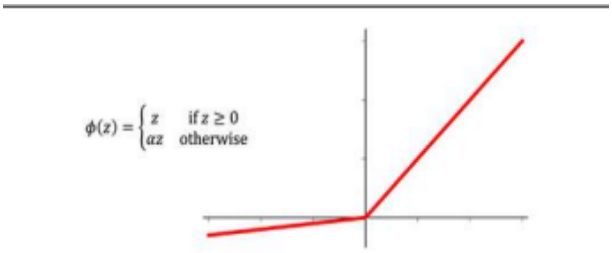


3 pav. $ReLU(x)$ funkcijos grafikas [17]

- *LeakyReLU* aktyvacijos funkcija

$$LeakyReLU(k, x) = \begin{cases} x, x \geq 0 \\ k \cdot x, x < 0 \end{cases} \tag{2.23}$$

  $LeakyReLU(k, x)$ padeda apskaičiuoti gradientų reikšmes, kurios galėtų būti ir teigiamos, ir neigiamos reikšmės. Tik veikia tokiu pagrindu, jeigu x yra didesnis arba lygus 0, tada grąžinamas modelio įvesties vektorius x. Kitu atveju, grąžinama x reikšmė, sudauginta su funkcijos krypties koeficientu k, kuris gali būti bet kokio tipo ir dydžio skaičius $[-\infty, \infty]$ 4 pav.

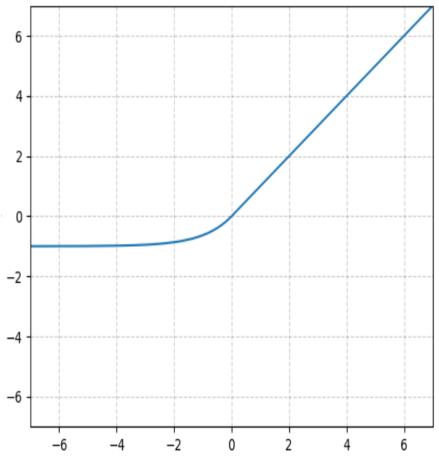


4 pav. $LeakyReLU(x)$ funkcijos grafikas [21]

- *ELU* aktyvacijos funkcija

$$ELU(x) = \begin{cases} x, x > 0 \\ \alpha(\exp(x) - 1), x \leq 0 \end{cases} \tag{2.24}$$

Jeigu funkcijos parametras x yra didesnis už 0, tuomet, x reikšmė yra nekeičiama. Kitu atveju, jeigu x reikšmė yra mažesnė arba lygi 0, tuomet, apskaičiuoja x reikšmės eksponentė pagal α parametrą. α krypties koeficientas, kurį funkcijos naudotojas gali apsibrėžti. α reikšmė priklauso intervalui $[-\infty, \infty]$.



5 pav. $ELU(x)$ funkcijos grafikas [16]

2.4. Optimizavimo algoritmai

Darbe naudojamas gradientų skaičiavimo optimizavimo algoritmas *Adam*. Reikalingus surasti linijinių transformacijų svorius, kurie yra reikalingi pagal duotą duomenų rinkinį x, skaičiuoti gilaus mokymo modelių latentinių kintamųjų aproksimacijas z [5].

Įvestis: γ (mokymosi dažnis 0,01)

Įvestis: $\beta_1, \beta_2 \in [0, 1)$ (Eksponentės mokymosi dažniai, kurie inicializuojami reikšmėmis β_1, β_2. Algoritmo autoriai rekomenduoja reikšmes naudoti $\beta_1 = 0,9$ ir $\beta_2 = 0,999$)

Įvestis: $f(\theta)$ tam tikro modelio netekties funkcija, kuri turi parametrų aibę θ

Įvestis: θ_0, pradinis modelio parametrų vektorius, kuris initializuotas atsitiktiniais skaičiais arba 1.

Įvestis: ϵ, vartotojo parenkamas pokyčio reikšmė. Dažniausiai naudojama reikšmė 10^{-8}.

$m_0 \leftarrow 0$ (Initializuota pirma vektoriaus momentinė reikšmė)

$v_0 \leftarrow 0$ (Initializuotas antra vektoriaus momentinė reikšmė)

$t \leftarrow 0$ (Pradinis laiko žingsnis)

for t = 1 to t_{max} **do**

$g_t \leftarrow \nabla_\theta f_t(\theta_{t-1})$

$m_t \leftarrow \beta_1 * m_{t-1} + (1 - \beta_1) * g_t$

$v_t \leftarrow \beta_2 * v_{t-1} + (1 - \beta_2) * g_t^2$

$\hat{m}_t \leftarrow m_t / (1 - \beta_1^t)$

$\hat{v}_t \leftarrow v_t / (1 - \beta_2^t)$

$\theta_t \leftarrow \theta_{t-1} - \gamma * \hat{m}_t / (\sqrt{\hat{v}_t}) + \epsilon$

end for

Norint, kad veiktų tinkamai *Adam* algoritmas (2 algoritmas), reikia vartotojui nurodyti gerą algoritmo mokymosi dažnio reikšmę γ (η). Reikšmė parenkama eksperimentiniu būdu žvelgiant pagal modelio, kuriam yra taikomas parametrų optimizavimo algoritmas, rezultatus. Keičiamas dažnis iki tol, kol vartotojui atrodys, kad modelis sukūrė tinkamus rezultatus. Modelio rezultatai yra įvertinami pagal vartotojo pasirinktas metrikas (W_1 (*wasserstein* atstumas, diagramas ir pan.)). Paprastai, kai bandoma atrasti tinkamą mokymosi dažnį, startuojama nuo žemiausios reikšmės 0,01 ir vėl mažinama arba didinama žiūrint pagal modelio sukurtus rezultatus.

Algoritmas veikia tam tikrame kiekyje iteracijų, kurios apribrežiamos t_{max}. Tai yra vartotojo nurodytas epochų skaičius, skirtas treniruoti gilaus mašininio mokymo modelį. Nurodžius t_{max}, apskaičiuojami vektorių momentinės reikšmės m_t ir v_t, pagal kurias vėliau atnaujinama θ_t modelio parametrų reikšmių aibę atimant pagal apskaičiuotą vertę išraiška $\gamma * \hat{m}_t / (\sqrt{\hat{v}_t}) + \epsilon$.

2.5. Paketo normalizavimas

Metodas, kuris yra taikomas pagreitinti gilaus mokymo modelio apmokymo procesą ir mažinti latentinės erdvės z aproksimacijos reikšmes, kurios gaunasi labai didelės. Modelio sluoksnių vektorių reikšmės, surastos pagal tiesinės transformacijos metodą yra transformuojamos pagal paketo normalizavimo metodą (*anglų k. batch normalization*), kuris naudoja vidurkį μ, standartinį nuokrypį σ ir parametrus γ ir β.

Metodo formulės, kurios yra reikalingos μ_B, σ_B ir y radimui.

$$\mu_B = \frac{1}{m}\sum_{i=1}^{m} x_i \tag{2.25}$$

$$\sigma_B^2 = \frac{1}{m}\sum_{i=1}^{m}(x_i - \mu_B)^2 \tag{2.26}$$

$$y = \frac{x_i - \mu_B}{\sqrt{\sigma_B^2 + \epsilon}} \tag{2.27}$$

m - įvesties vektoriaus ilgis. ϵ parametras, kuris yra vartotojo parenkamas ir reikšmė priklauso intervalui $[-\infty, \infty]$.

3. Duomenų rinkinys

Duomenų generavimo eksperimentams bus naudojamas aukštos energijos, fizikinių, keturių leptonų dalelių modelio duomenų rinkinys [1]. Duomenų rinkinys yra *.csv* failas, kurį sudaro 20,000 eilučių ir 20 stulpelių. Failas yra sugeneruotas *CERN* organizacijos mokslininkų, taikant *Monte Carlo* metodu paremtą naujų duomenų generatorių. Rinkinio stulpeliuose yra pateikta fizikinės *Z Higgs* bozono dalelių formavimo ir branduolio skylimo įvykiuose dalyvaujančių dalelių koordinačių taškų, energijos, dalelių kategorijos informacija:

- *E1*, *E2*, *E3*, *E4* - stulpeliai vaizduoja fizikiniuose įvykiuose dalyvaujančių, miuonų, protonų bei elektronų dalelių išskirtą energiją. Aprašoma skaičiais, kurie priklauso realiųjų skaičių aibei $\mathbb{R}$. Fizikoje leptonų, protonų ir elektronų dalelių energijos vienetai yra elektro voltai.

- *pdg1*, *pdg2*, *pdg3*, *pdg4* - stulpeliai yra nurodantys fizikiniuose įvykiuose dalyvaujančių dalelių kategorinius tipus. Fizikinių dalelių tipai stulpeliuose yra nusakomi fiksuotais, sveikaisiais skaičiais. 13 ir -13 žymi miuonus ir antimiuonus. -11 ir 11 žymi elektronus ir protonus.

- p1x, p2x, p3x, p4x, p1y, p2y, p3y, p4y, p1z, p2z, p3z, p4z - stulpeliai apibūdinantys fizikiniuose *Z Higgs* bozono dalelių formavimo ir branduolio skylimo dalyvaujančių dalelių įvykių koordinates. Koordinačių reikšmės priklauso realiųjų skaičių aibei $\mathbb{R}$.

Prieš naudojant keturių leptonų duomenų rinkinį naujų duomenų generavimo eksperimentams, pagal tam tikrą giliojo mokymo modelį, taikomas retėjimo vaizdavimo metodas pasirinktam duomenų rinkiniui (*anglų k. sparse encoding*). Kiekviename duomenų rinkinio įraše yra reikšmės, kurios informuoja kokios dalelės dalyvauja tam tikrame fizikiniame įvykyje ir 0, randantys, kokios dalelės nedalyvauja.

Duoto duomenų rinkinio dimensija yra padidinama iš 20 į 32. Idėja metodo taikymo yra pagreitinti modelių mokymo procesą, nes modelių mokymo cikluose, ne visi duoto duomenų rinkinio įrašai yra transformuojami pagal linijines transformacijas [22]. Kiekviename įraše yra dalis nulinių reikšmių ir dalis nenulinių reikšmių. Duomenų rinkinyje vaizduojama kiekviename duomenų rinkinio įrašas, kuris apibūdiną 8 dalelių dalyvaujančiuose 4 leptonų įvykiuose. Kokios reikšmės kiekviename įraše yra užpildomas, *sparse* algoritmas sprendžia pagal fizikinių dalelių tipus *pdg1*, *pdg2*, *pdg3* ir *pdg4*. Tarkime *pdg1 = -11* ir *pdg2 = 11*, tuomet užpildomi duomenų rinkinio įrašai susieti su elektronų ir protonų susidūrimo įvykiu. Jeigu *pdg3 = 13* ir *pdg4 = -13*, tuomet miuono ir atimiuono atributai duomenų rinkinio įraše yra užpildomi. Naujų stulpelių pavadinimai:

1. *el1_neg_E, el1_neg_px, el1_neg_py, el1_neg_pz* - fizikiniame įvykyje dalyvaujančio pirmo elektrono energijos *el1_neg_E* ir koordinačių x, y ir z informacija.

2. *el2_neg_E, el2_neg_px, el2_neg_py, el1_neg_pz* - fizikiniame įvykyje dalyvaujančio antro elektrono energijos *el2_neg_E* ir koordinačių x, y ir z informacija.

3. *el1_pos_E, el1_pos_px, el1_pos_py, el1_pos_pz* - fizikiniame įvykyje dalyvaujančio pirmo pozitrono energijos *el1_pos_E* ir koordinačių x, y ir z informacija.

4. *el2_pos_E, el2_pos_px, el2_pos_py, el1_pos_pz* - fizikiniame įvykyje dalyvaujančio antro pozitrono energijos *el2_pos_E* ir koordinačių x, y ir z informacija..

5. *mu1_neg_E, mu1_neg_px, mu1_neg_py, mu1_neg_pz* - fizikiniame įvykyje dalyvaujančio pirmo miuono energijos *mu1_neg_E* ir koordinačių x, y ir z informacija.

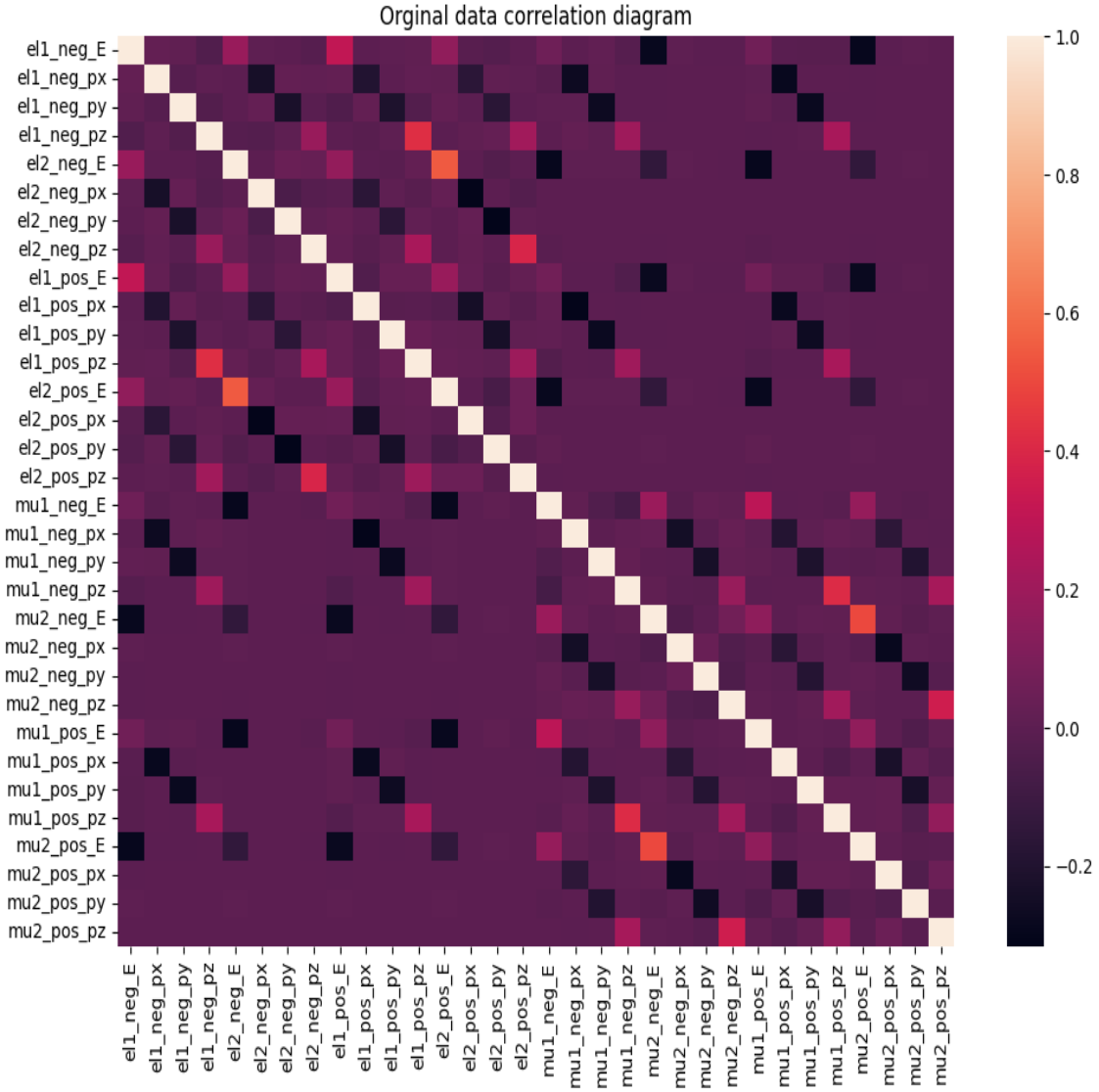
6. *mu2_neg_E, mu2_neg_px, mu2_neg_py, mu2_neg_pz* - fizikiniame įvykyje dalyvaujančio antro miuono energijos *mu2_neg_E* ir koordinačių x, y ir z informacija.

7. *mu1_pos_E, mu1_pos_px, mu1_pos_py, mu1_pos_pz* - fizikiniame įvykyje dalyvaujančio pirmo antimiuono energijos *mu1_pos_E* ir koordinačių x, y ir z informacija.

8. *mu2_pos_E, mu2_pos_px, mu2_pos_py, mu2_pos_pz* - fizikiniame įvykyje dalyvaujančio antro antimiuono energijos *mu2_pos_E* ir koordinačių x, y ir z informacija.

3.1. Duomenų rinkinio savybės

Apskaičiavus koreliacijos matricą pagal Pearsono metodą, galima pamatyti, kaip duomenų rinkinio stulpelių reikšmės koreliuoja tarpusavyje. Diagramoje yra rodoma, kad kuo labiau šviesesnė spalva, tuo labiau stipresnė koreliacija duomenų 6 pav. Tamsesnė spalvos diagramoje rodo silpnesnę duomenų koreliaciją.



6 pav. *Monte carlo Pearsono* duomenų koreliacija

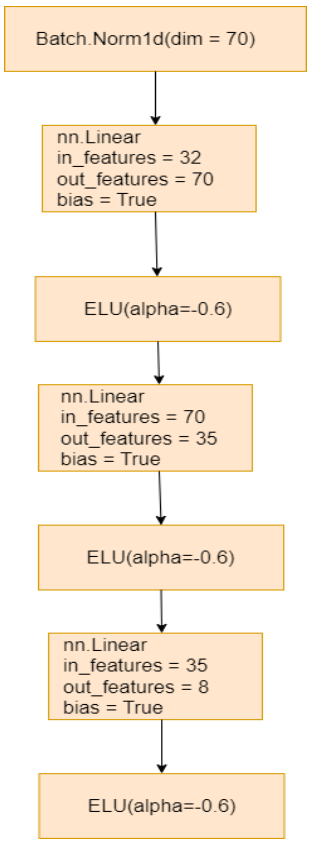
Ištyrinėjus koreliacijos diagramą, matosi, duomenų koreliacija yra nepastovi. Gerai įsižiūrėjus į diagramą, matosi, kad tamsioje violetinėje spalvoje, matosi neryškiai skirtingos koreliacijų reikšmės. Reiškia, kad duomenys duotame duomenų rinkinyje, fizikinių dalelių įvykiai yra vienas nuo kito nepriklausomi ir neturintys kokio nors sąryšio.

4. Duomenų generavimo eksperimentai

Šioje dalyje yra aprašomi atlikti gilaus mokymo modelių eksperimentai. Atliekami duomenų generavimo eksperimentai naudojant 4 leptonų duomenų rinkinį ir *VAE*, *GAN* bei normalizuojančius srautų modelius. Kiekvienas modelis buvo apmokomas pagal duoto duomenų rinkinio mokymo aibę ir visą duomenų aibę. Sugeneravus naujus fizikinius dalelių duomenis, išanalizuojami nauji sugeneruoti duomenys pagal *Frechet Inception* atstumą, *Wasserstein* atstumą, invariantinės masės metrika M_{inv}, bendrą dalelių energiją E ir dalelių greičius p_T. Tiriant rezultatus, siekiama nustatyti, ar duomenys yra tiksliai apskaičiuojami, įgauna Gauso skirstinio savybes ir panašūs į pradinius *Monte Carlo* duomenis. Atlikus naujų duomenų analizę, ištyrinėjama ar mašininio mokymo modeliai nenaudoja daug procesoriaus skaičiavimo laiko ir operatyvios *RAM* atminties generuojant naujus fizikinius aukštos energijos duomenis.

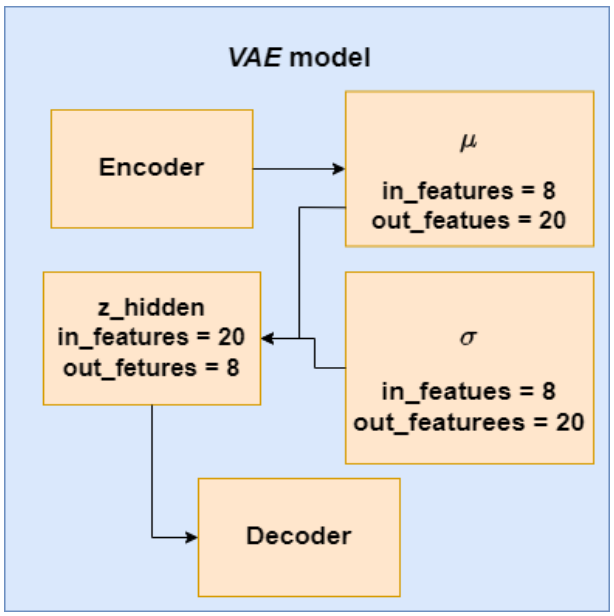
4.1. *VAE* eksperimentai

Realizuotoje *VAE* architektūroje, enkoderio sluoksnį sudaro trys tiesinės transformacijos sluoksniai `nn.Linear(in_features, out_features)`. Pirmame tiesinėje transformacijoje yra apibrėžta, kad 32 įvesties matrica (`in_features`) bus transformuojama į 70 dimensijos matricą (*out_features*). Antroje tiesinės transformacijos sluoksnyje yra nurodyta, kad pirmo sluoksnio apskaičiuota matrica, kurios dimensija yra 70 bus transformuojama į 35 dimensijos matricą. Trečiame sluoksnyje iš 35 matricos bus transformuojama į 8 dimensijos matricą. Enkoderio architektūra naudoja vieną *BatchNorm1d* paketo normalizacijos sluoksnį, kuris yra taikomas papildomai transformuoti duomenis, tam kad galima būtų naudoti aukštesnes mokymo dažnio koeficiento reikšmes modelio apmokymui, neprarandant reikšmių stabilumo. Norint, kad *VAE* modeliui pavyktu sugeneruoti naujus duomenis, kurie turi netiesiškumo bruožų, naudojami trys *ELU* aktyvacijos sluoksniai enkoderyje. Atlikus visus skaičiavimus enkoderio sluoksnyje, apskaičiuojamos latentinės erdvės Z reikšmės pagal surastą vidurkį μ ir σ.



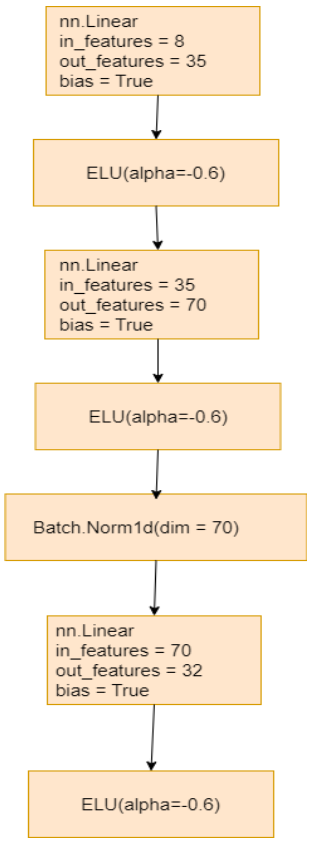
7 pav. Realizuota VAE enkoderio architektūra

Modeliui suradus enkoderio duomenų aproksmacijų reikšmių aibių matricą, iš jos yra surandamos vidurkio μ matrica, kovariacijos matrica σ ir latentinės erdvės Z reikšmių aibė. Norint surasti μ ir σ matricas yra naudojamos *nn.Linear* tiesinės transformacijos funkcijos. μ sluoksnis, tiesinės transformacijos metu, naudoja 8 dimensijos įvesties bruožų matricą ir apskaičiuoja 20 dimensijos vidurkių matricą. σ sluoksnis kaip ir vidurkių matricą, naudoja 20 dimensijos įvesties matricą ir apskaičiuoja 8 dimensijos σ matricą. *VAE* modelis apskaičiavęs šiuos duomenis, panaudoja Z latentinės erdvės reikšmių skaičiavimui, kuriame naudojama *nn.Linear* tiesinė transformacijos funkcija. Z sluoksnis surandamas naudojant 8 dimensijos matricą, gautą pagal vidurkių μ ir kovariacijos matricą σ(8 pav.).



8 pav. Detalesnė VAE architektūros informacija

VAE dekoderio modelio struktūra yra panašios struktūros kaip ir enkoderio modelio. Tik latentinės erdvės z duomenys yra transformuojami atvirkštinę tvarką. Tik paketo normalizavimo metodas yra taikomas prieš paskutinį *ELU* sluoksnį (9 pav.).



9 pav. VAE dekoderio architektūra

Tokia yra įgyvendinta architektūrą, nes kad *VAE* modelis praradęs truputi tikslumo, galėtų beveik tiksliai sugeneruoti duomenis. Testuojant skirtingas *VAE* architektūrų implementacijas, pavyko nustatyti, kad didėjant linijinių transformacijų sluoksnių kiekiui ir naudojamų linijinių išvesties

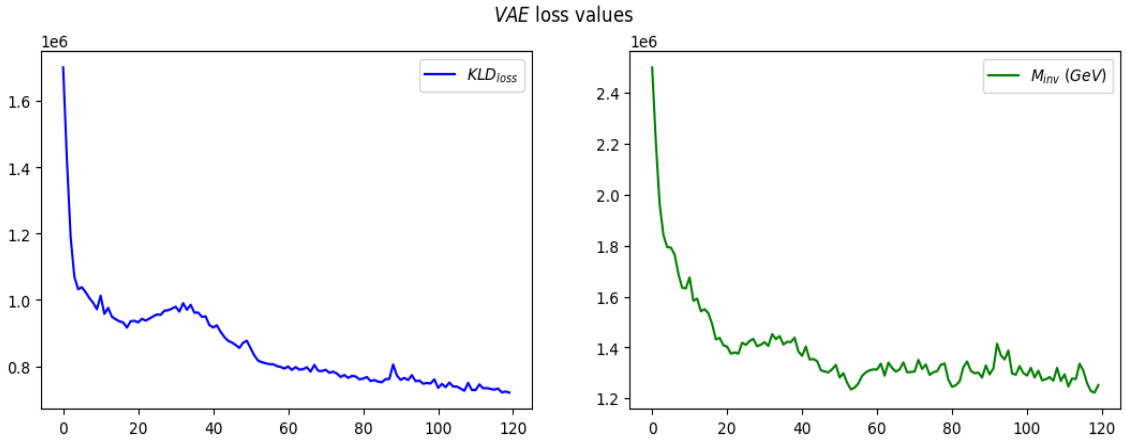
dimensijų (neuronai), ilgėja kompiuteriaus procesoriaus skaičiavimo laikas, naudojama daugiau kompiuterinės atminties ir gali gautis mažiau tikslesni sugeneruoti nauji duomenys. Pavyzdžiui, keliuose stulpeliuose maža dispersija naujame duomenų rinkinyje, naujame duomenų rinkinyje duomenys mažiau kaip atsitiktiniai skaičiai ir pan. Norint išsaugoti skaičių atsitiknumo bruožą, naudoja *ELU* aktyvacijos funkcija.

VAE modelio treniravimo metu, naudojamas 120 epochų ciklas, nes straipsnio autoriams [24] pavyko sugeneruoti naujus fizikinius dalelių duomenis taikant 120 epochų kiekį. *VAE* modelio mokymas vyko pagal mokymo aibę ir visą aibę. *VAE* modelis buvo treniruojamas skirtingomis mokymosi dažniais η. Kiekviena kartą apmokius modelį *VAE* buvo įvertinamas *FID* atstumas. Treniravimo ciklas vyko pagal *Adam* algoritmą. Pavyko nustatyti, kad geriausias mokymo dažnio koeficiento η lygus 10^{-4}, nes gaunasi mažesnis *FID* atstumas.

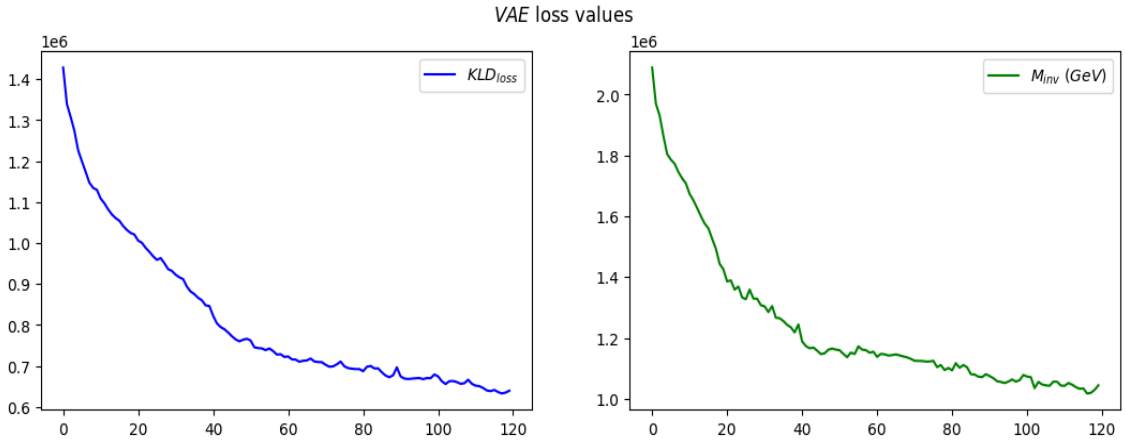
1 lentelė. *VAE* modelio treniravimo *FID* rezultatai pagal skirtingus mokymosi dažnio koeficientus η

η	*FID* atstumas
10^{-5}	$5,185 * 10^{-11}$
10^{-4}	$8,98 * 10^{-11}$
$1,5^{-4}$	$5,99 * 10^{-11}$

VAE modelio mokymo procese, pavyko gauti mažėjančią *VAE* modelio netekties funkcijos vertę, kuri skaičiuojama pagal *KL* divergenciją ir vidutinę kvadratinę paklaidą. Gavosi, kad netekties funkcija beveik yra logaritmiškai mažėjanti. Apmokant *VAE* modelį, buvo dar vienas tikslas surasti invariantinės masės netekties funkcijos mažesnę vertę, kuri yra skaičiuojama pagal fizikinę, invariantinės masės formulę. Pasiekus minimumą ties 120 epochos, turėtų *VAE* mokėti sugeneruoti tinkamus naujus fizikinius dalelių duomenis pagal 4 leptonų duomenų rinkinio mokymo arba visą aibę. Išbandžius abi aibes, gaunasi panašūs netekties funkcijų rezultatai. Abu grafikai beveik logaritmiškai mažėja. Tik pagal mokymo aibę, 120 epochoje pasiektas minimumas, o pagal visą aibę, pradeda šiek tiek didėti (10 pav. ir 11 pav.).

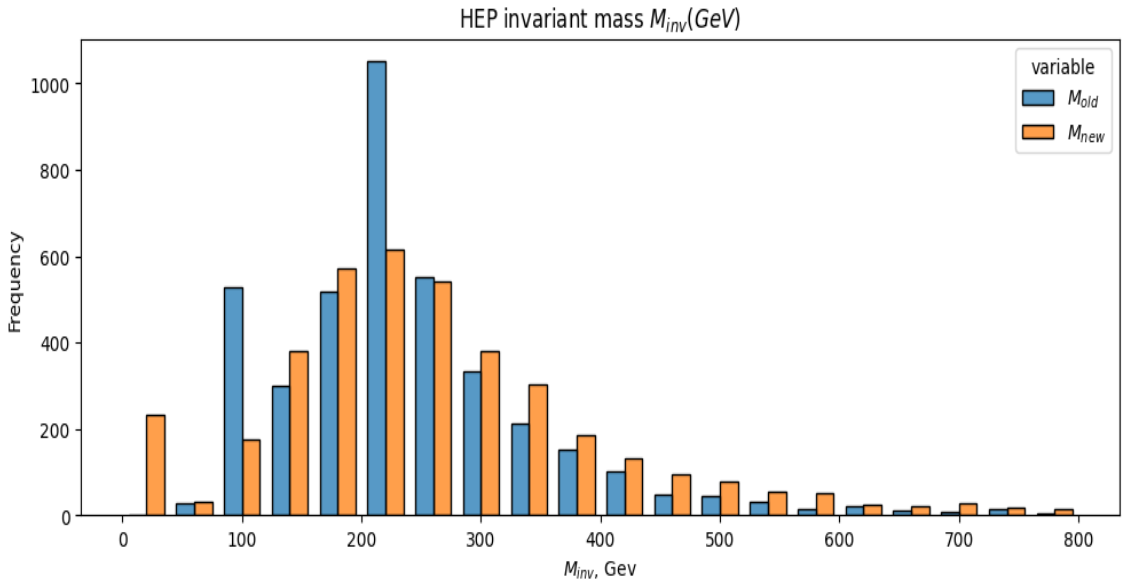


10 pav. VAE netekties funkcijos reikšmės 120 epochų treniravimo cikle pagal mokymo aibę



11 pav. VAE netekties funkcijos reikšmės 120 epochų treniravimo cikle pagal visą aibę.

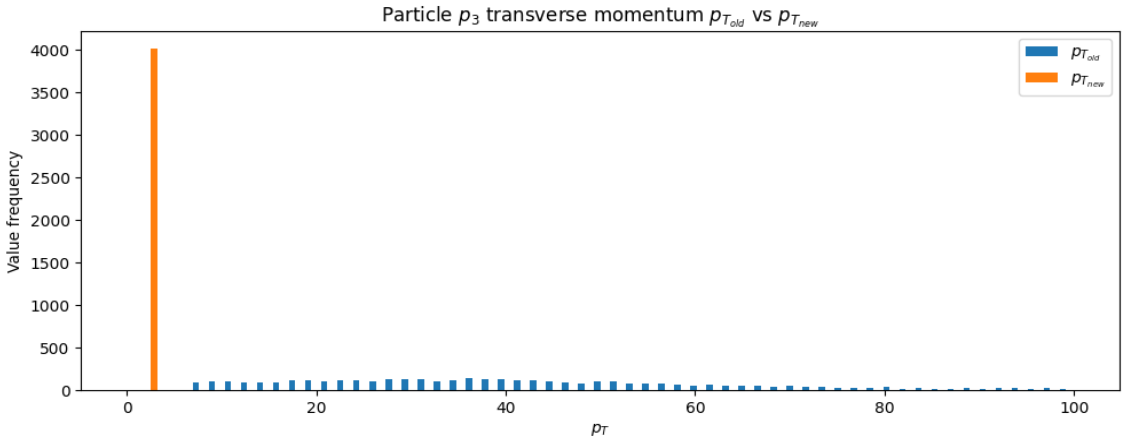
Atlikus *VAE* modelio apmokymo procesą, buvo vykdomas dalelių duomenų generavimo eksperimentas. Atrastas modelio aproksimacijos μ ir σ buvo panaudotos Z paslėpto, latentinės erdvės sluoksnio duomenų aproksimacijos skaičiavimams. Suradus Z aibės reikšmių aibę, panaudojamos *VAE* dekoderio sluoksnio duomenų generavimo automatiniame procese. Įvykdžius *VAE* duomenų generavimą, pavyko gauti duomenis, kurių invariantinė masė M_{inv} turėtų Gauso skirstinio savybių. Invariantinė masė, kuri yra apskaičiuota pagal sugeneruotus fizikinius duomenis yra žymima oranžine spalva, o invariantinė masė, kuri apskaičiuota pagal duotus duomenis, pažymėta mėlyna spalva 12 pav. Duotų duomenų invariantinės masės reikšmės koncentruojasi labiau ties 200 skaičiaus. Sugeneruotų duomenų, pagal *VAE* modelį, invariantinės masės reikšmės koncentruojasi prie reikšmės $X \in [200, 300]$ Tai reiškia, kad sugeneruotų duomenų invariantinės masės M_{inv} reikšmės yra beveik arti reikiamo centro. Apskaičiavus *Wasserstein-1* atstumą pagal invariantinę masę, kuri yra rasta pagal duotus duomenis ir sugeneruotus, gaunasi $W_1 = 101.853$.



12 pav. Invariantinės masės M_{inv} histograma pagal dalelių koordinates ir energijas. Duomenis yra gauti taikant mokymo aibę

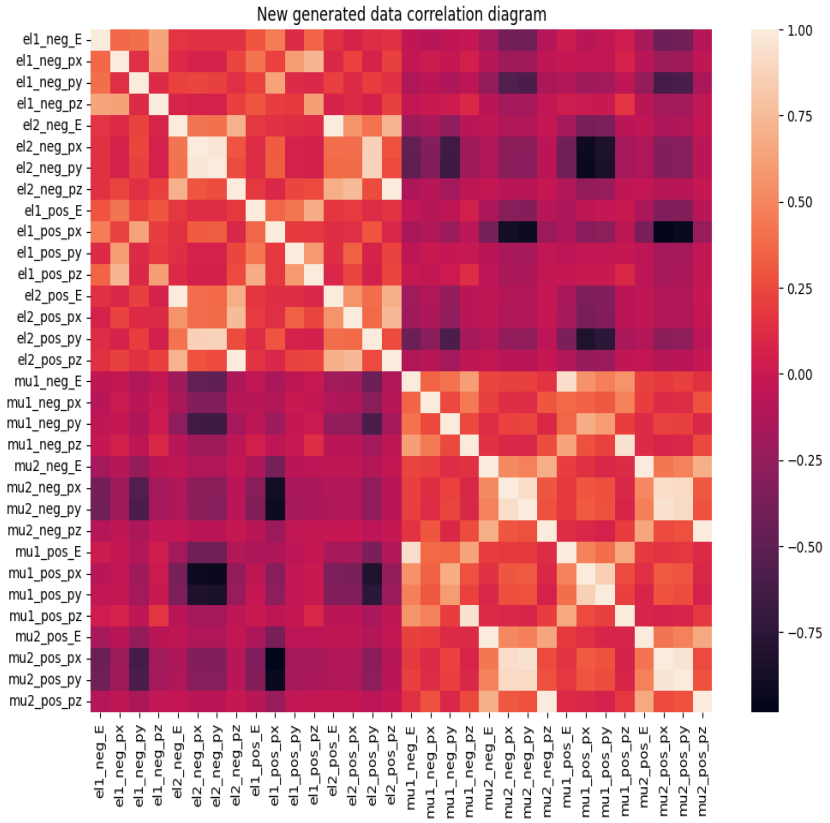
Apskaičiavus dalelių greičių p_T koordinates, gavosi blogi rezultatai. *VAE* modeliui pavyko sugeneruoti kiekvienos dalelės energiją, bet negalėjo sugeneruoti tinkamai x ir y dalelių koordinates.

Grafike 13 pav. yra vaizduojama dalelių greičių histograma.



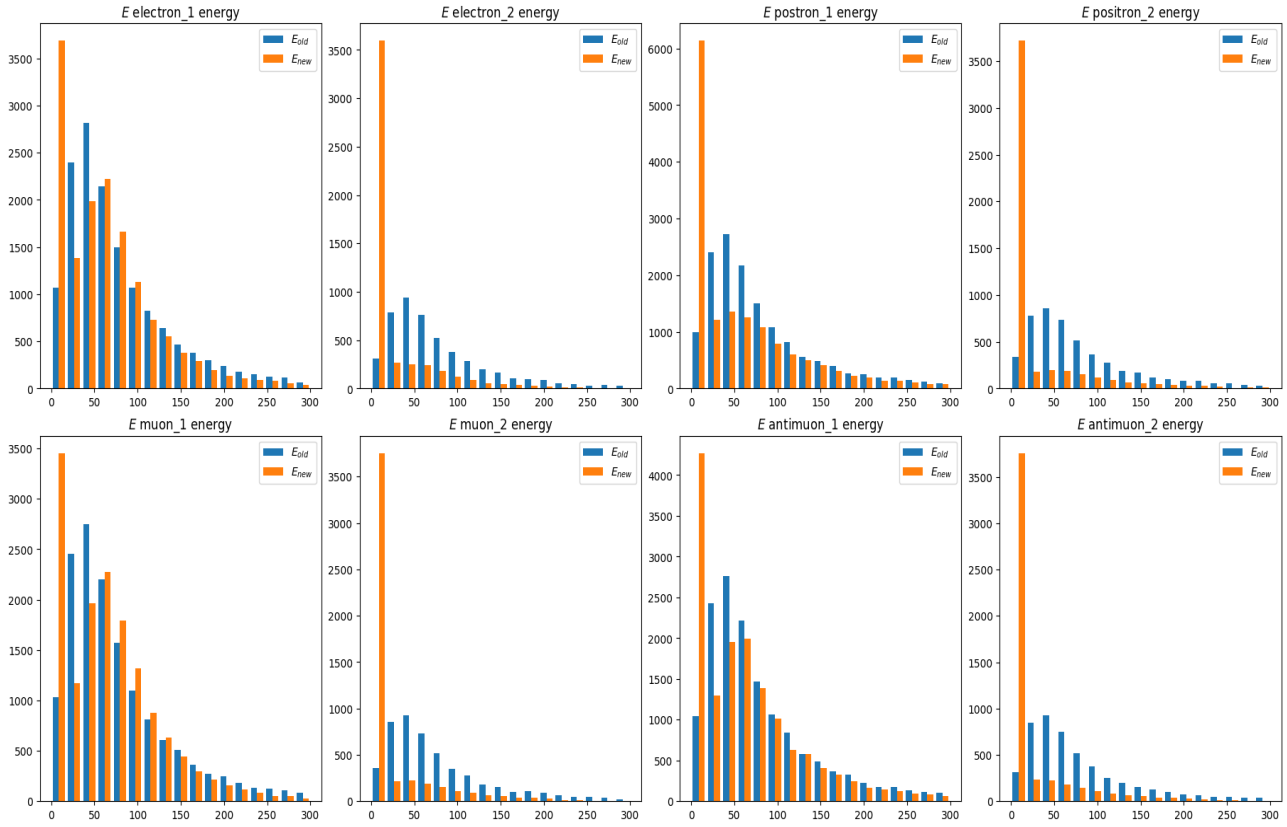
13 pav. Sugeneruotų duomenų ir duotų duomenų p_T dalelių greičio palyginimas

Palyginus M_{inv} mases, buvo dar lyginami duotų duomenų ir sugeneruotų duomenų dalelių greičiai p_T. Dalelių greičiai buvo apskaičiuojami pagal du pabėgusius elektronus iš įvykusio dalelės branduolio formavimo įvykio. Diagramoje apskaičiuota dalelių greičio aibę p_T pagal duotus duomenis žymima $p_{T_{old}}$. Dalelių greičių aibę, kuri rasta pagal sugeneruotus duomenis, žymima $p_{T_{new}}$. Analizuojant rastų duomenų histogramą, gaunasi, kad abiejų aibių savybės skiriasi, apskaičiuota dalelių greičių aibė pagal sugeneruotus duomenis, turi daug reikšmių kurios būtų netoli 0. Dalelių greičiai $p_{T_{old}}$ turi didesnių reikšmių, kurios priklausytų invervalui [0, 100]. Apskaičiuotų duomenų *Wasserstein-1* atstumas yra lygus 43.984.



14 pav. Sugeneruotų duomenų Pearsono koreliacijos diagrama

Jeigu bandytume sugeneruoti Pearsono koreliacijos diagramą pagal duomenis, kurie yra sugeneruoti taikant *VAE* modelį, matytume, jog koreliacijos diagrama šiek tiek skiriasi nuo Pearsono koreliacijos diagramos, apskaičiuotos pagal pradinius duomenis. Sugeneruotų 4 leptonų duomenų koreliacijos diagrama, turi daug reikšmių, kurios turi turi stiprią koreliaciją, pažymėta raudona spalva netoli diagramos įstrižainės. Dar naujų dalelių duomenys turi reikšmių, kurios turi silpna (juoda spalva) arba vidutinę (violetinę) koreliaciją. Tai reiškia, kad koreliacijos diagrama gali būti pakankamai informatyvi ir turi stulpelių, kurių reikšmės skiriasi viena nuo kitos. Lyginant abi koreliacijos diagramas, apskaičiuota frobenijaus normos M_f reikšmė lygi 13,429.



15 pav. Pagal mokymo aibę *VAE* modeliu sugeneruotų ir pradinių duomenų E energijos

Jeigu mėgintume lyginti duotų duomenų E_{old} ir sugeneruotų duomenų E_{new} apskaičiuotas dalelių energijas, galime pastebėti, kad grafikai yra skiriasi (16 pav.), kai kurie grafikai įgauna kreivę, kuri panaši į gauso kreivę, bet dauguma turi daug reikšmių kurios yra arti link 0. Grafikų centrai yra beveik panašūs kai kuriuose grafikuose. Matosi, kad *VAE* modeliui nepavyksta ir kai kurių dalelių energijas tinkamai sugeneruoti.

2 lentelė. *VAE* sugeneruotų ir duotų duomenų μ ir σ skirtumai

Vertės pavadinimas	μ	σ
Duotų *HEP* duomenų	11,289	9.321
Sugeneruotų *HEP* duomenų	63,095	37.239

Peržiūrėjus bendrai abiejų duomenų rinkinių vidurkių μ ir σ skirtumus, galime pastebėti, jog sugeneruotų duomenų, pagal *VAE* modelį vidurkis μ ir σ gaunasi beveik panašus. Pradiniuose

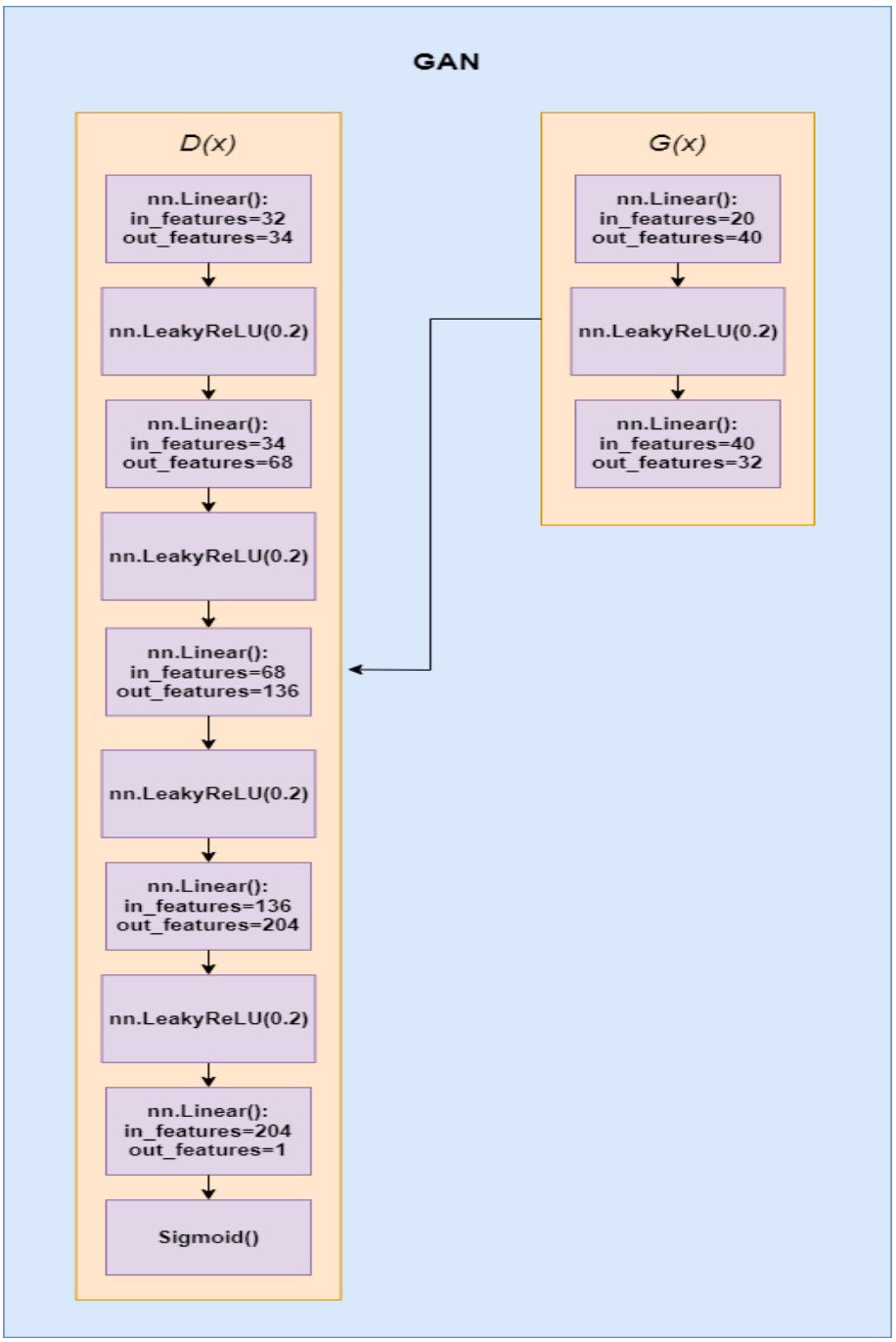
duomenyse standartinis nuokrypis σ yra nutolęs nuo vidurkio μ ir sugeneruotuose naujuose fiziki-niuose duomenyse standartinis nuokrypis σ yra nutolęs nuo vidurkio μ. Tiktais pradinių duomenų vidurkis yra lygus 63,095, o sugeneruotų duomenų vidurkis yra 37,239. Gavosi, kad daugiau reikšmių yra nutolę pradiniuose duomenyse negu naujuose duomenyse (2 lentelė). Dėl to, daugiau analitinės informacijos apie fizikines galima pamatyti kolkas duotuose duomenyse.

4.2. *GAN* duomenų generavimo eksperimentai

Atlikti naujų, aukštos energijos, fizikinių dalelių duomenų generavimo eksperimentai taikant *GAN* modelį. Modelis turi realizuotą tam tikrą architektūrą, leidžiančią generuoti naujus duomenis. Architektūrą sudaro diskriminatoriaus ir generatoriaus modeliai. Įgyvendinant *GAN* modelio architektūrą buvo stengiamasi padaryti, taip, kad *GAN* modelis galėtų nenaudojant daug įvesties ir išvesties neuronų, sugeneruoti naujus fizikinius duomenis analitiniams moksliniams darbams. Įgyvendinus *GAN* architektūrą, atlikti modelio apmokymo ir duomenų generavimo procesai. Modelio apmokymo metu, pagal *Binary Cross Entropy (BCE)* ir M_{inv} , buvo žiūrima ar diskriminatoriaus netekties funkcijos reikšmė didėja ir generatoriaus reikšmė mažėja.

Darbe realizuotas *GAN* modelio diskriminatoriaus $D(x)$ ir generatoriaus $G(x)$ modeliai. Realizuota $D(x)$ architektūra, kur būtų sudėtingesnės struktūros už generatoriaus $G(x)$. Diskriminatoriuje, buvo siekiama sukurti 204 dimensijos tikimybių masyvą, iš kurio pataptų į 1 dimensijos tikimybinį masyvą. Kiekviename tiesinės transformacijos *nn.Linear()* sluoksnyje dvigubai didėja neuronų kiekis (*out_features*). Prieš paskutiniame diskriminaroriaus sluoksnyje pasiekus reikiamą maksimumą, neuronų kiekis yra sumažinamas iki 1. Vėliau pagal apskaičiuotas tikimybines reikšmes yra surandama *sigmoid()* funkcijos tikimybinė reikšmė. Modelis suradęs *sigmoid* reikšmę, gali ją panaudoti *Binary Cross Entropy(BCE)* skaičiavime.

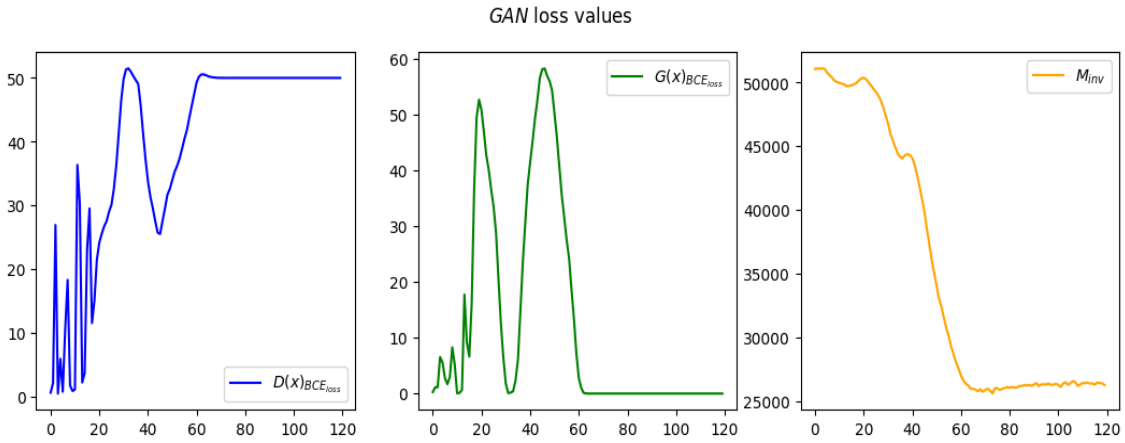
GAN modelio generatoriaus metodas yra labiau paprastesnės struktūros. Užteko realizuoti trejų sluoksnių generatoriaus architektūrą, kurią sudaro linijinę transformaciją *nn.Linear()*, turinčią įvestyje 20 neuronų (bruožų, *in_features*) ir 40 neuronų (bruožų, *out_features*) išvestyje bei sluoksnį, turintį aktyvacijos funkciją *nn.LeakyReLU()* ir linijinės transformacijos sluoksnį *nn.Linear()*, kurio įvestyje 40 neuronų ir išvestyje 32 neuronų.



16 pav. Naudota *GAN* modelio architektūra

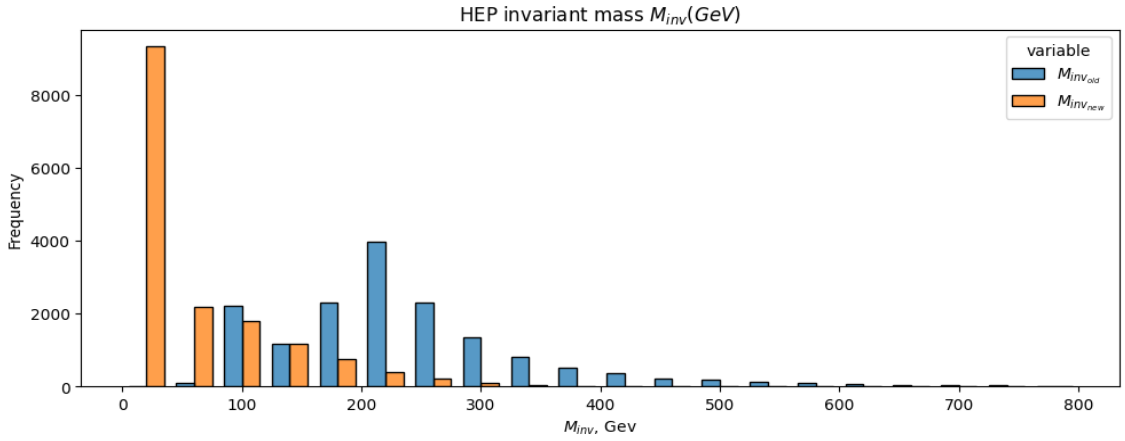
$D(x)$ padarytas sudėtingesnės architektūros, nes jeigu modelyje yra didesnis tiesinių sluoksnių kiekis, pavyksta tam tiksliau apskaičiuoti klasifikavimo tikimybes. $G(x)$ struktūra buvo padaryta labiau paprastesnė, nes $D(x)$ yra sudėtingesnės struktūros ir jeigu $G(x)$ modelis irgi būtų sudėtingesnės struktūros, abu modeliai daugiau virtualios atminties naudotų ir daugiau procesoriaus skaičiavimo laiko. Dėl greitaveikos, padaryta paprastenės generatoriaus architektūra.

Apmokant *GAN* modelį pastebėta, kad gaunasi, diskriminatoriaus metodo *BCE* reikšmė nemažėja, o didėja. Generatoriaus *BCE* reikšmė labiau mažėjanti 17 pav. Modelio apmokymas buvo įvykdytas su 120 epochų treniravimo ciklu. Treniravime tikslas buvo, kad *GAN* modelio $D(x)$ netekties funkcijos reikšmė būtų aukštesnė, jeigu generatoriaus $G(x)$ reikšmės yra panašios į treniravimo aibės duomenis ir generatoriaus reikšmė $G(x)$ būtų mažesnė, jeigu metodui pavyksta sugeneruoti naujus duomenis, panašius į treniravimo aibę. Modelio treniravime gavosi, kad diskriminatoriaus reikšmė didėja ir įsisotina ties 50, o generatoriaus netekties reikšmė įsisotina ties 0. M_{inv} invariantinės masės netekties reikšmės eksponentiškai mažėja kaip *VAE* modelio treniravimo **??** skyrelyje.



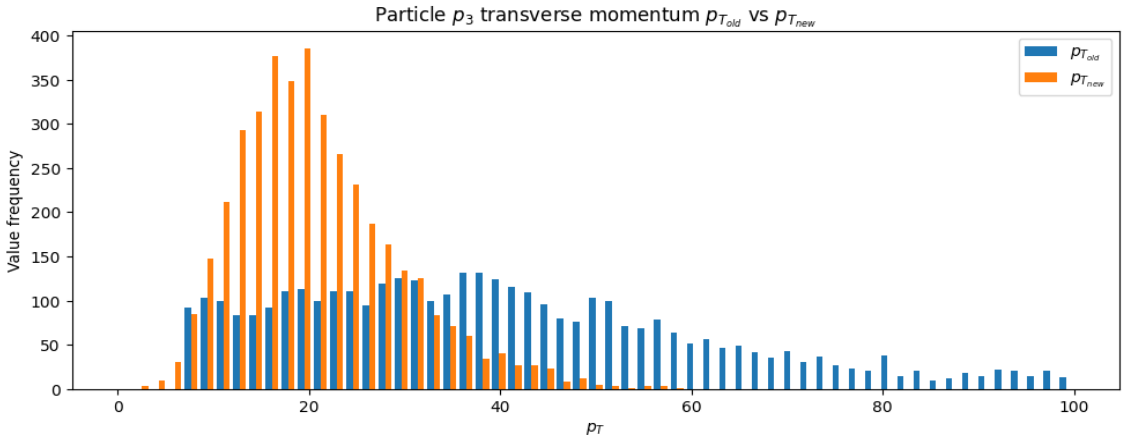
17 pav. *GAN* netekties *BCE* ir M_{inv} vertės

Pritaikius fizikines metrikas pastebėta, kad gaunasi, *GAN* modelio invariantinė masės M_{new} grafikas ir duotų duomenų invariantinės masės grafiko M_{old}, W_1 atstumas lygiu 51,248. Susidaro didesnis invariantinės masės skirstinių skirtumas negu *VAE* modelyje. 18 pav.



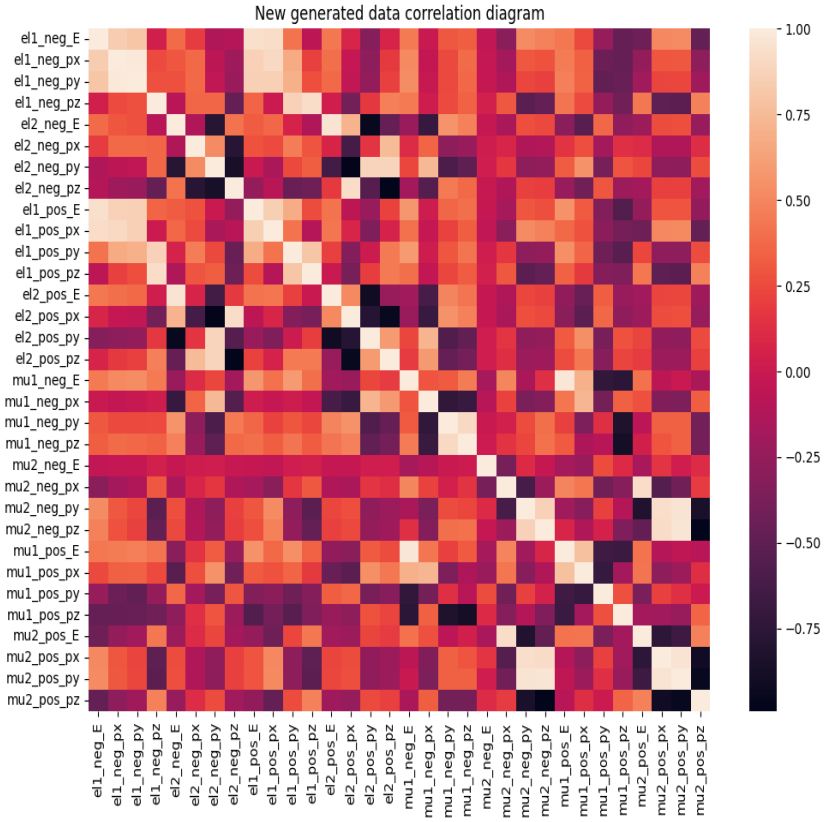
18 pav. M_{inv} skirstiniai sugeneruotų $M_{inv_{new}}$ ir duotų duomenų $M_{inv_{old}}$

Abiejų grafikų reikšmių centrai beveik sutampantys. $M_{inv_{old}}$ grafiko, gauso kreivės tipo, histogramų centras susidaro prie 200 vertės, sugeneruotų duomenų histogramos centras susidaro intervale $[200, 300]$. Sugeneruotų duomenų invariantinės masės reikšmių variacija pakankamai didelė, nes centre esantis standartinis nuokrypis sugeneruotų duomenų invariantinės masės $M_{inv_{new}}$ yra aukštesnis už duotų duomenų invariantinės masės $M_{inv_{old}}$.



19 pav. *GAN* modelio sugeneruotų duomenų ir duotų duomenų p_T dalelių greičio palyginimas

Analizuojant pabėgusių elektronų greičio grafikus $p_{T_{old}}$ ir $p_{T_{new}}$ *GAN* modeliu irgi gavosi geresni dalelių greičio rezultatai. Modeliui pavyko sėkmingai sugeneruoti dalelių įvykių koordinates, iš kurių sėkmingai pavyko apskaičiuoti dalelių greičius p_T. *GAN* modelio sugeneruotuose duomenyse, p_T W_1 atstumas yra mažesnis už *VAE* modelio.



20 pav. Sugeneruotų duomenų Pearsono koreliacijos diagrama

Analizuojant sugeneruotų duomenų Pearsono koreliacijos diagramą, sugeneruotų duomenų stulpeliai arba koreliuoja arba ne. Kai kuriose diagramos dalyse, reikšmių koreliacija kartojasi. Yra didelė reikšmių koreliacijos įvairovė. Reiškia, kad duomenys yra labiau nepastovūs, triukšmingi, turintys tam tikrų pasikartojimų. Koreliacijos frobenijaus normos reikšmė M_f yra lygi 13,626. Bendras koreliacijų dydžių skirtumas yra didesnis lyginant su *VAE* modelio frobenijaus

normos reikšme(14 pav). Tai reiškia, kad duomenų koreliacijos savybės daugiau skiriasi negu *VAE* modelio koreliacijos diagramos.

3 lentelė. *GAN* modeliu sugeneruotų ir duotų duomenų μ ir σ skirtumai

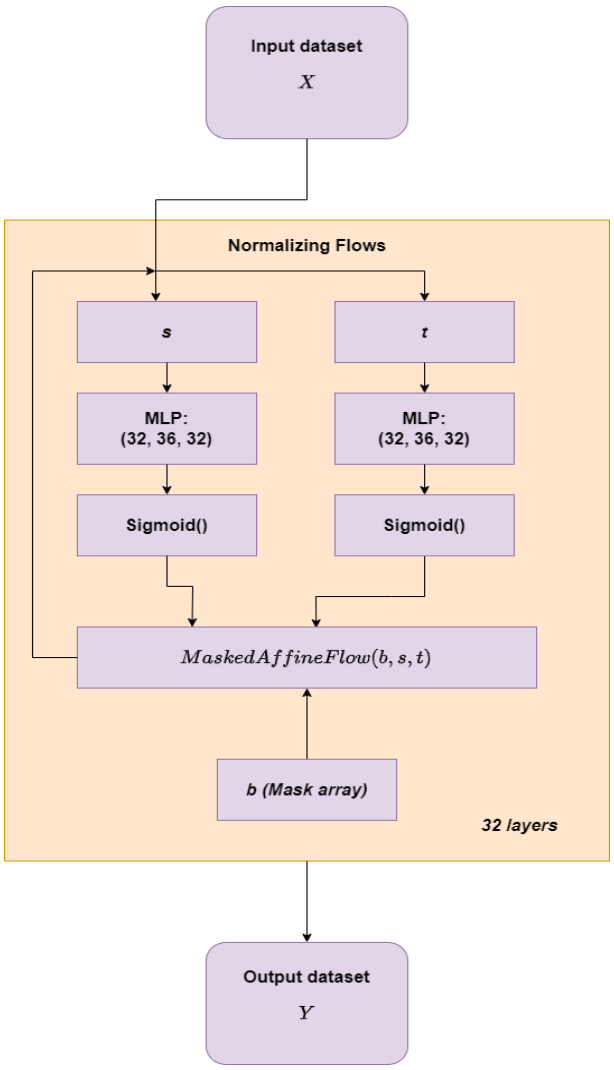
Vertės pavadinimas	μ	σ
Duotų *HEP* duomenų	11,289	0.962
Sugeneruotų *HEP* duomenų	63,095	13.65

Tikrinant vidurkį μ ir σ *GAN* modelio sugeneruotų duomenų, galima pamatyti, jog sugeneruotuose duomenyse yra mažesnė variacija duomenų (3 lentelė). Naujų sugeneruotų duomenų σ standartinis nuokrypis nutuolęs nuo vidurkio μ, panašiai kaip *VAE* modelyje, tik *GAN* sugeneruotuose duomenyse μ ir σ yra mažesnis už *VAE* modeliu sugeneruotų duomenų. Tai reiškia, kad *VAE* sukūrė labiau didesnį duomenų informatyvumą, o *GAN* modelis daugiau sumažino informatyvumą, bet variacija tarp duomenų, *GAN* modeliui pavyko išlaikyti, kad būtų panašus į duotų duomenų. Naujuose duomenyse yra atstiktiniu reikšmių savybių.

4.3. Normalizavimo srautų duomenų generavimas

Parengtas naujų *HEP* duomenų generavimo eksperimentas naudojant normalizavimo srautų modelį. Realizuota modelio architektūra, naudoja normalizavimo srautų maskuotus, autoregresyvius srautų sluoksnius (*Masked Affine Flow*) ir daugiasluoksnius perceptrono sluoksnius. Atliktas modelio apmokymas pagal *Adam* gradientų optimizavimo algoritmą, *KL* divergencijos atstumą ir invariantinės masės M_{inv} netekties funkcijų reikšmes. Sugeneruoti duomenys būdavo paduodami į modelio *KL* divergavimo atstumo netekties funkciją, kuri naujus sugeneruotus duomenis palygina su duotais duomenimis.

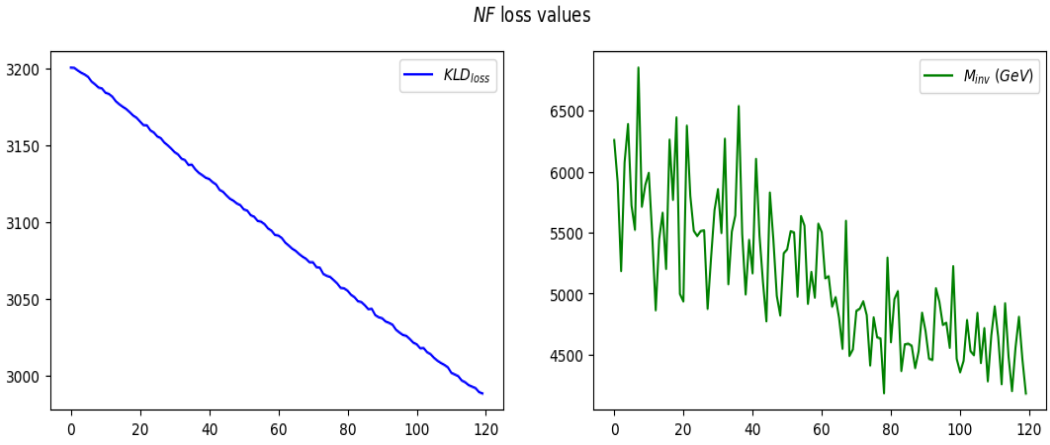
Modelio architektūrą sudaro du daugiasluoksniai perceptrono (*MLP*) sluoksniai, kurie yra panaudojami maskuotame sraute (*anglų k. Masked Affine Flow*) (21 pav.). Normalizavimo srauto apmokymui yra parinkti 32 srauto sluoksniai. *MLP* sluoksnius sudaro trys svorių matricos, kurios 32. 36 ir 32 dimensijų. Dar ir sluoksniuose yra naudojama netiesinė aktyvacijos *ReLU()* funkcija tam kad būtų matricų reikšmių sandaugoje netiesiškumo bruožai. Kai duotų duomenų aproksimacijos yra apskaičiuotos iš matricų svorių daugybos ir netiesinių *ReLU()* funkcijų, pritaikoma tiesinė aktyvacijos funkcija - sigmoidė. Įvesties matricos aproksimacijos yra skaičiuojama dviejose vienodose *MLP* sluoksniuose pažymėtais s ir t. Vėliau modelio architektūroje *MLP* sluoksniai naudojami maskuotame sraute, kuriame yra atliekamos diferencialinės transformacijos. Atlikus diferencialines aproksimacijų transformacijas, gautos reikšmės vėl paduodamos *MLP* sluoksniams s ir t. Toks procesas modelio architektūroje daromas 32 kartus. Šiuo principu, pavyko gauti duomenys, kurie įgautų norimų duomenų rinkinio savybių.



21 pav. Implementuota normalizavimo srautų architektūra, taikant *pytorch API normflows* biblioteką

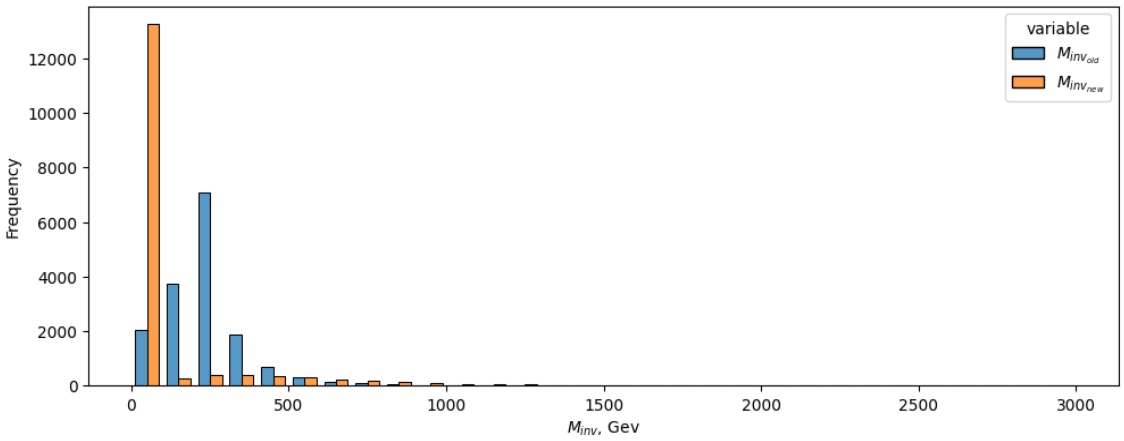
4.3.1. Modelio treniravimas ir naujų duomenų generavimas

Apmokant normalizavimo srautų modelį pagal duotus *Z Higgs bozono* duomenis, buvo skaičiuojamas *KL* divergavimo funkcijos reikšmė ir invariantinės masės M_{inv} netekties funkcijos reikšmė. Modelio apmokymas buvo vykdomas 120 epochų. Darant modelio apmokymo eksperimentus, siekiama sumažinti netekties funkcijos reikšmes. Modelis buvo apmokytas 120, 160 ir 100 epochomis ir nustatyta, kad kuo didesnis epochų skaičius tuo labiau ilgėja normalizavimo srautų modelio apmokymo proceso laikas. Parinktas 120 skaičius suteikę padaryti taip, kad modelis greičiau, geriau išmoktų generuoti naujus fizikinius duomenis. Gavosi, kad netekties funkcijos *KL* divergencijos reikšmės tiesiškai mažėja iki tam tikro minimumo, o invariantinės M_{inv} funkcijos mažėja, bet nestabiliai.



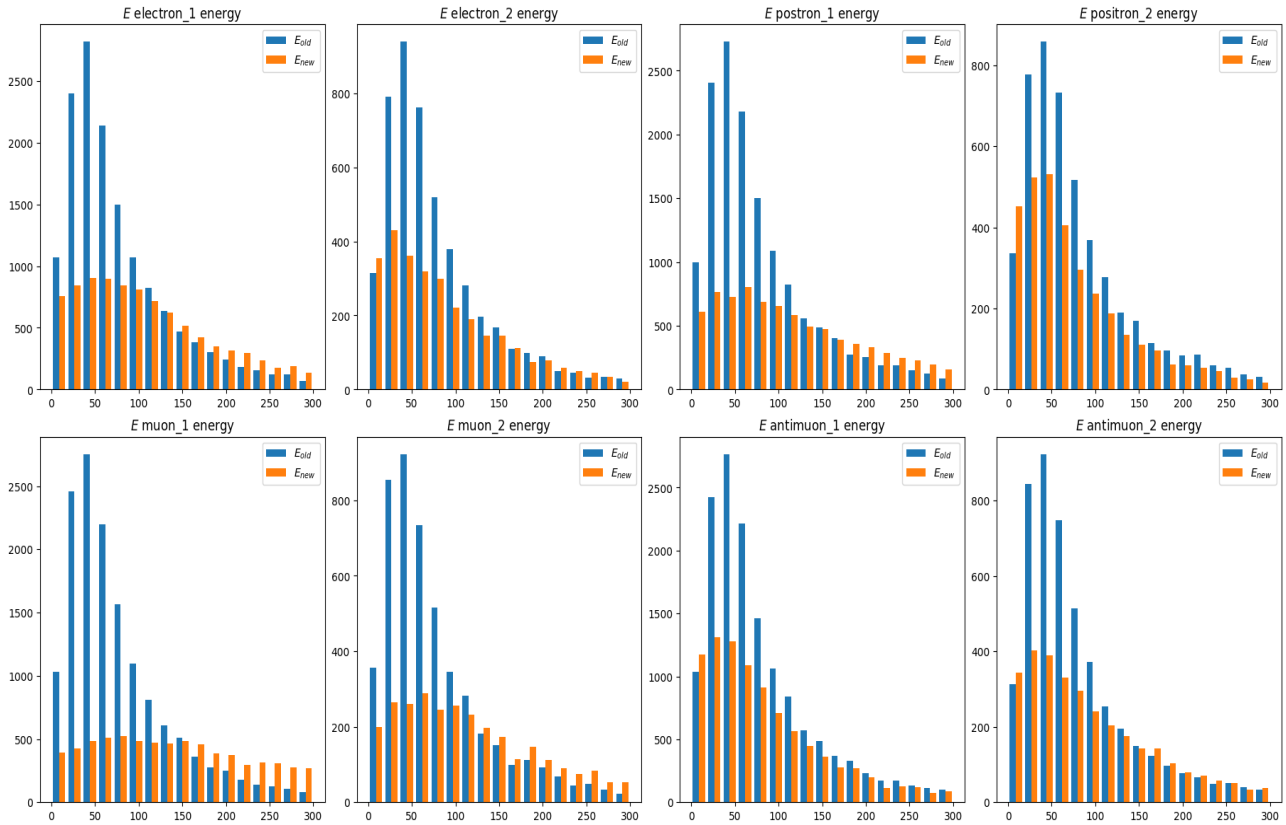
22 pav. Normalizavimo srautų modelio netekties funkcijų reikšmės

Pagal šiuo grafikus gavosi taip, kad modeliui yra sudėtinga apmokyti pagal invariatinės masės netekties funkciją. Tai gali gautis, naujuose, sugeneruotuose aukštos energijos dalelių fizikiniuose duomenyse, tam tikrų netikslumų skaičiuojant kokią nors analitinę informaciją.



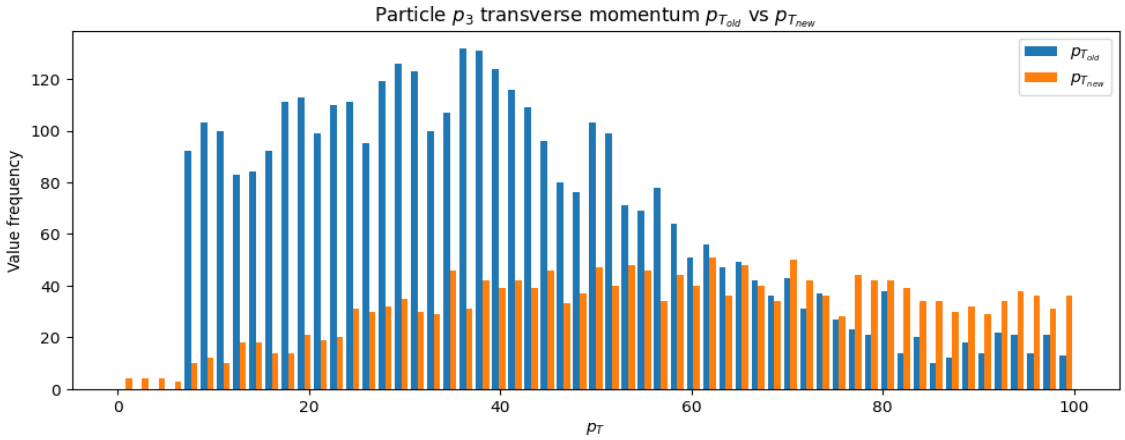
23 pav. M_{inv} skirstiniai sugeneruotų $M_{inv_{new}}$ ir duotų duomenų $M_{inv_{old}}$

Lyginant sugeneruotų duomenų invariantinę masę $M_{inv_{new}}$ su $M_{inv_{old}}$ (23 pav.), gaunasi, kad normalizavimo srauto modeliui pavyksta sunkiau aproksimuoti reikšmes taip kad pavyktų tikslingiau apskaičiuoti invariantinės masę naujų dalelių duomenų. Abiejų invariantinės masės skirstinių W_1 atstumas gaunasi lygus 200,599. Palyginus šią W_1 reikšmę su kitų modelių invariantinės masės rezultatais, *VAE* modelis sugeneruoja šiek tiek tikslingiau invariantinės masės naujus fizikinius dalelių duomenis su atstumu W_1 = *71,636*.



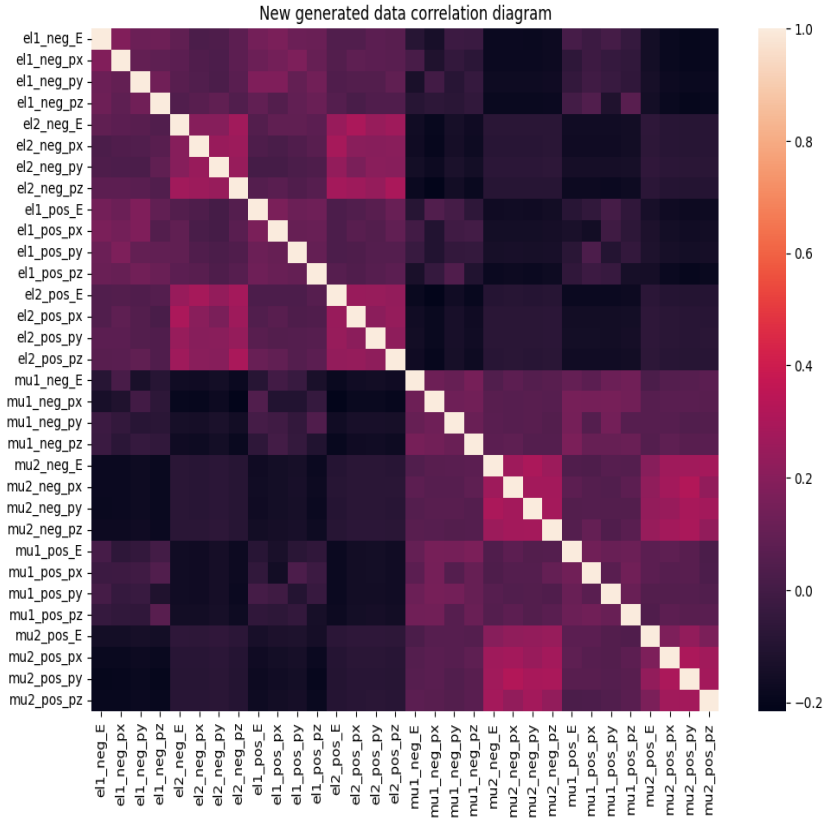
24 pav. Sugeneruotų ir duotų fizikinių duomenų E energijos

Patikrinant sugeneruotų naujų fizikinių dalelių energines vertes (24 pav.), gaunasi, kad normalizavimo srautų modeliui pavyksta geriau aproksimuoti dalelių energijos vertes. Beveik visuose grafikuose susidaro gauso pavidalo histogramų kreivės, susikoncentravusios ant tam tikro energijos vidurkio. Standartiniai nuokrypiai yra beveik visuose grafikuose dideli. Dalelių duomenys turi pakankamai informatyvius energijos duomenis.



25 pav. Normalizavimo srautų modelio sugeneruotų duomenų ir duotų duomenų p_T dalelių greičio palyginimas

Pabėgusių dalelių greičio histogramos gaunasi beveik panašios 25 pav. Tik standartinis nuokrypis p_T naujų duomenų diagramoje susidaro mažesnis ir nėra aukštas kaip duotų duomenų p_T standartinis nuokrypis. $p_{T_{old}}$ ir $p_{T_{new}}$ W_1 atstumas yra lygus 44.953.



26 pav. Sugeneruotų duomenų Pearsono koreliacijos diagrama

Sugeneruotų duomenų, pagal normalizavimo srautų modelį, susidaro nevienodą koreliacijos diagramoje violetinę spalvą. Atsiranda nevienoda koreliacija tarp naujo, sugeneruoto duomenų rinkinio kintamųjų. Reiškia, kad naujo, duomenų rinkinio stulpeliai yra mažiau vienas nuo kito priklausomi. Galima bus atrasti kokios nors naudingos analitinės informacijos. Tikslas yra sugeneruoti duomenis informatyvius, kuriuose koreliacija yra nevienoda 26 pav. Frobenijaus norma M_f pagal naujų duomenų koreliacijos diagramą ir duotų duomenų koreliacijos diagramą, gaunasi $M_f = 4.994$. Koreliacijos skirtumas yra beveik panašus abiejų sugeneruotų duomenų. *GAN* ir *VAE* modelių frobenijaus normos atstumai yra dideli. *VAE* modelio frobenijaus norma lygi $9,402$, o *VAE* modelio frobenijaus norma lygi 13.626.

4.4. Rezultatų ir greitaveikos palyginimas

Šiame skyriuje bendrai apžvelgiami visų tyrinėtų gilaus mokymo modelio duomenų generavimo ir veikimo greitaveikos rezultatus. Atliekant aukštos energijos duomenų generavimo eksperimentus, siekiama buvo pritaikyti gilaus mokymo modelį, kuris galėtų sugeneruoti kokybiškus aukštos energijos fizikinius dalelių duomenis. Atlikus duomenų generavimo eksperimentą, pritaikomos fizikinės ir statistinės metrikos, kurios parodo, ar naujų sugeneruotų duomenų savybės yra arti pasirinkto duomenų rinkinio. Taikomos metrikos, skirtos įvertinti modelių duomenų generavimo greitaveiką. Tikrinama, kiek reikia laiko apmokyti modelį pagal naujus duotus duomenis ir sugeneruoti naujus fizikinius duomenis.

4 lentelė. Modelių *FID* atstumo ir laiko rezultatai

Modelio pavadinimas	*FID* atstumas pagal duotus ir naujus *HEP* duomenis
VAE	$8,1989 * 10^8$
GAN	$3,527 * 10^7$
NF	$2,942 * 10^9$

Viena iš metrikų, kuri buvo taikoma įvertinti modelių sugeneruotų duomenų kokybė buvo *Frechet Inception* atstumas. Metrika skirta analizuoti dviejų duotų matricų vidurkio ir standartinio nuokrypio skirtumas. Išsiaiškinti ar dviejų matricų skirstiniai yra panašūs. Peržiūrėjus modelių rezultatus 4 lentelės rezultatus, galime matyti, kad tiksliausiai duomenis generuoja priešingų tinklų modelis *GAN*. Analizuojant *GAN* skyrelio rezultatus, galime pastebėti, kad modeliui pavyksta sugeneruoti duomenis pagal kuriuos paskaičiavus dalelių greičius, gaunasi mažesnis W_1 (*Wasserstein*) svoris. Analizuojant invariantinės masės skirtumą, matome, kad skirtumas yra šiek tiek didesnis. Sugeneruotuose naujuose duomenyse pagal *NF* (Normalizuojantys srautai), lyginant su *VAE* ir *GAN* frobinaujaus norma yra mažiausia. Dar ir analizuojant visų sugeneruotų duomenų dispersijas, pastebėta, kad *NF* modeliui pavyko visų stulpelių duomenis aproksimuoti ir kiekviename stulpelyje yra dideli standartiniai nuokrypiai kaip duotame *Monte Carlo* būdu sugeneruotame duomenų rinkinyje. Tai reiškia, kad *NF* modeliui pavyko padaryti tokius duomenis, kurie būtų informatyvūs, duomenų reikšmės kai kurios yra toliau nuo vidurkio.

5 lentelė. Modelių dalelių skirstinių *FID* atstumai pagal duotus ir sugeneruotus *HEP* duomenis

Dalelė	*VAE FID*	*GAN FID*	*NF FID*
Elektronas 1	$1,125 * 10^8$	$4,821 * 10^6$	$7,37304913884 * 10^8$
Elektronas 2	$5,131 * 10^6$	$8,336 * 10^5$	$5,5013945853 * 10^7$
Positronas 1	$2,531 * 10^6$	$4,37 * 10^6$	$2,62221448581 * 10^8$
Positronas 2	$7,75 * 10^8$	$9,75 * 10^5$	$2,62221448581 * 10^8$
Miuonas 1	$1,33 * 10^8$	$3,27 * 10^6$	$5,22445337265 * 10^8$
Miuonas 2	$6,501 * 10^6$	$1,192 * 10^6$	$8,405698758 * 10^7$
Antimiuonas 1	$1,51 * 10^8$	$4,075447537 * 10^6$	$8,49649257899 * 10^8$
Antimiuonas 2	$4,601 * 10^6$	$1,04 * 10^6$	$1,91 * 10^8$

Analizuojant visų dalelių duomenų skirstinių skirtumus pagal duotus ir sugeneruotus *HEP* duomenis, gavosi, kad *GAN* modelis geriausiai sugeneravo naujus duomenis 5 lentelė. Dauguma modelio sugeneruotų duomenų skirstinių *FID* atstumai yra maži lyginant su *VAE* ir *NF* modelio *FID* atstumais.

Kadangi darbe siekiama pasiekti, kad norima turėti kiek įmanoma visų metrikų mažiausias reikšmes, *NF* modelio frobenijaus normą gaunasi pati mažiausią, lyginat su *VAE* ir *GAN* modelio. Normalizavimo srautų modelis gautųsi kol kas tiksliausias (6 lentelė).

6 lentelė. Modelių M_f normos skaičiavimo rezultatai

Modelio pavadinimas	M_f (frobenijaus norma)
VAE	8,94
GAN	11,183
NF	5.002

Atliekant modelių apmokymo ir naujų duomenų generavimo eksperimentus, buvo siekiama ne tik įvertinti sugeneruotų duomenų tikslumą pagal tam tikras pasirinktas metrikas, bet ir dar nustatyti ar taikomi modeliai efektyviai naudoja kompiuterinę virtualią atmintį ir skaičiavimo laiką generuojant naujus fizikinius duomenis. *Monte Carlo* generatoriai naudoja *VEGAS* algoritmą, kuris generuoja fizikinių duomenų įvykius lėtai, kai kiekvieno įvykio sukūrimas trunka apie 10 min.

7 lentelė. Modelių skaičiavimo laikas ir virtualios atminties *GPU* naudojimas apmokymo ir duomenų generavimo metu

Modelis	*GPU* išnaudojo (*MB*)	Laikas *Google Colab*	Laikas superkompiuteryje
VAE	92.16	3 min.	24min 28s
GAN	499.712	2s	8s
NF	172.032	4min 30s	9min 1s

Pasirinkti modeliai duomenų generavimo tyrimams buvo paleidžiami *Google Colab* aplinkoje ir universiteto *HPC* superkompiuteryje. Paleidžiant modelių duomenų generavimo eksperimentus buvo tikranama, kiek išnaudojo *GPU* plokšties atminties procesų paskirstymų metu ir kiek laiko išnaudojo modelių apmokymo ir naujų duomenų generavimo metu. *GPU* naudojimas buvo tikrinamas naudotoje *Google Colab* aplinkoje. Modelių apmokymų ir naujų duomenų generavimo laikas buvo skaičiuojamas tiek *Google Colab*, tiek superkompiuterio aplinkoje.

Daugiausiai virtualios *GPU* išnaudojo *GAN* modelis. Reikia patobulinti modelio architektūrą, parinkti tinkamus hiperparametrus ir taikyti metodus, kurie patobulintų gradientų skaičiavimus, tam kad būtų mažiau naudojama virtualios atminties. Mažiausiai naudoja *GPU* atminties *VAE* modelis, bet paanalizavus gautus *VAE* modelio statistinius rezultatus pagal naujus sugeneruotus duomenis, modelis veikia efektyviai atminties naudojimo atžvilgiu, bet ne tiksliai skaičiuoja. *NF* (normalizavimo srautai) naudoja *172* MB. Palyginus su *VAE* ir *GAN* modelio *GPU* atminties naudojimo kiekiais, *NF* modelis naudoja nei per daug nei labai mažai. Analizuojant *NF* modelio veikimą pagal tokį virtualios atminties naudojimo kiekį, sugeneruoja naujus fizikinius duomenis

beveik tiksliai. *GAN* modelis naudoja daugiausiai virtualios *GPU* atminties, skaičiuojant diskriminaroriaus ir generatoriaus reikšmes sukūriant naujus fizikinius duomenis.

Naujų duomenų generavimo skaičiavimuose ilgiausiai trunka *VAE* modelis superkompiuteryje. Tai gali atsitikti dėl modelio architektūros, naudoja pakankamai daug operacijų transformuojant duotus duomenis ir dėl superkompiuterio aparatinės įrangos skaičiavimo ribojimų. Yra nepakankamai galingi *Google colab* paslaugai skirta aparatinė įrangą. *GAN* modeliai skaičiuoja greičiausiai generuojant naujus fizikinius duomenis, bet sukuria naujus fizikinius duomenis netiksliai. *NF* modelis pasiekia geresnį skaičiavimų greitį superkompiuteriuje, nes modelyje buvo geresnę architektūra ir hiperparametrai generuoti naujus fizikinius duomenis.

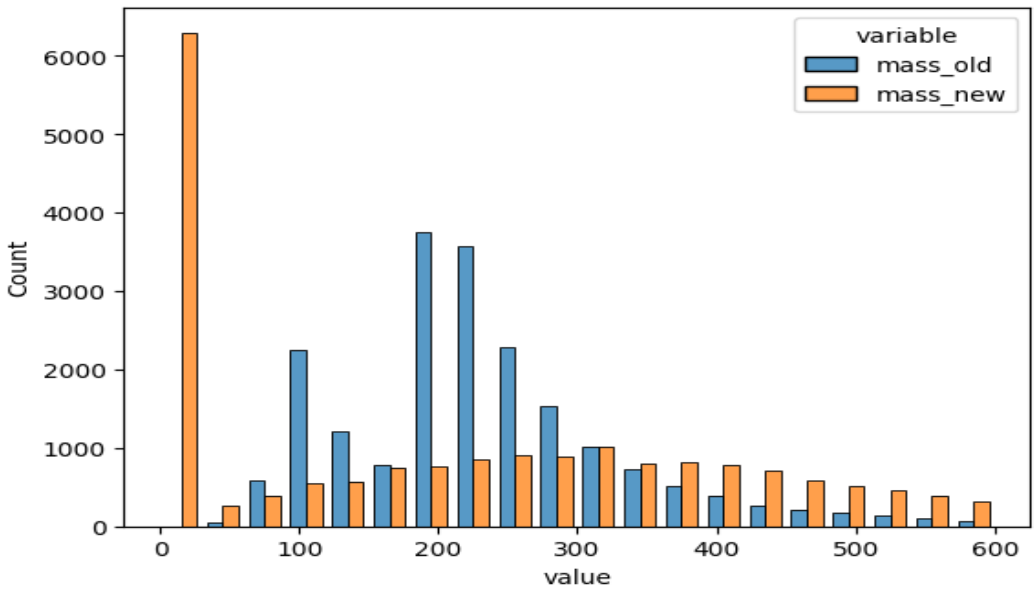
4.5. *HEP* duomenų generavimas pagal Gauso maišos modelį (*GMM*)

Atlikus $\hat{f}(x)$ funkcijos aproksimacijos eksperimentus pagal skirtingas atrastas hiperparametrų poras, buvo daromas duomenų generavimo eksperimentas pagal mokymo aibę X_{train}. Buvo paleidžiamas trejų iteracijų ciklas duomenų generavimo automatiniame scenarijuje. Kiekvienoje ciklo iteracijoje buvo iš naujo aproksimuojama *GMM* modelio generavimo funkcija $\hat{f}(x)$ pagal tam tikrą atrastą hiperparametrų porą, kurią sudaro *n_components* ir *covariance_type*. Ciklo iteracijos viduje, po atlikto aproksimacijos proceso, buvo sugeneruojami nauji fizikinių dalelių duomenys ir apskaičiuojama invariantinės masės $mass_{new}$ aibė. Apskaičiavus $mass_{new}$, buvo surandamas *Wasserstein* atstumas ir apskaičiuojama *Bray Curtis* nepanašumo įvertis pagal $mass_{old}$ ir $mass_{new}$ invariantinės masės aibes.

8 lentelė. Duomenų generavimo rezultatai pagal *GMM* modeli

n_components	covariance_type	Wasserstein	Bray Curtis
94	tied	117,2	44%
80	tied	108,75	43%
42	tied	107,35	43%

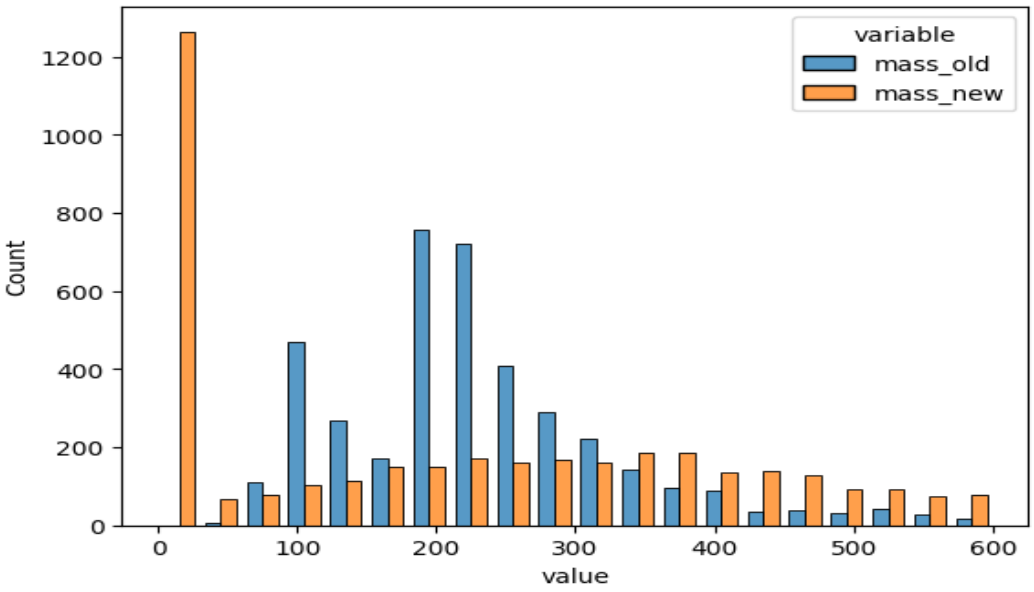
Lentelėje matome, kad didėjant *n_components* kintamajam, didėja *Wasserstein* ir *Bray Curtis* įvertis. Geriausias eksperimente rezultatas būtų lentelės trečias įrašas, kai *n_components = 42*, *Wasserstein* atstumas lygus 107,35 ir *Bray Curtis* atstumas lygus *43%*.



27 pav. Invariantinės masės m_{old} ir m_{new} palyginimas, kai yra naudojama mokymo aibė X_{train}

Diagramoje 27 pav. vaizduojamos invariantinių masių $mass_{old}$(mėlyna spalva) ir $mass_{new}$(oranžine spalva) histograma, kai *Wasserstein* atstumas lygus 107,35 ir *Bray Curtis* skirtumo atstumas lygus 43%. Histogramoje matosi pakankamai dideli skirtumai tarp masių aibių histogramų, nes eksperimentuose nustatyta, kad *GMM* modelis sugeneruoja duomenis pakankamai greitai, bet netiksliai. Apskaičiavus $mass_{new}$ aibę, joje gavosi daug klaidingų apskaičiuotų masės reikšmių. Skaičiuojant invariantinės masę, formulės(2.20 formulė) šaknies dalyje susidarė labai didelės reikšmės, iš

kurių ištraukus kvadratinę šaknį, *Python* scenarijuje gaunasi *NaN* vertės. 27 pav. $mass_new$ aibė įgavo 6909 *NaN* įverčių, kurie yra aptinkami 20000 ilgio $mass_{new}$ aibėje. Panašų rezultatą, taikant atrinktą geriausią hiperparametrų *n_components* ir *covariance_type* porą, galima pastebėti generuojant duomenis pagal testinę aibę X_{test}.

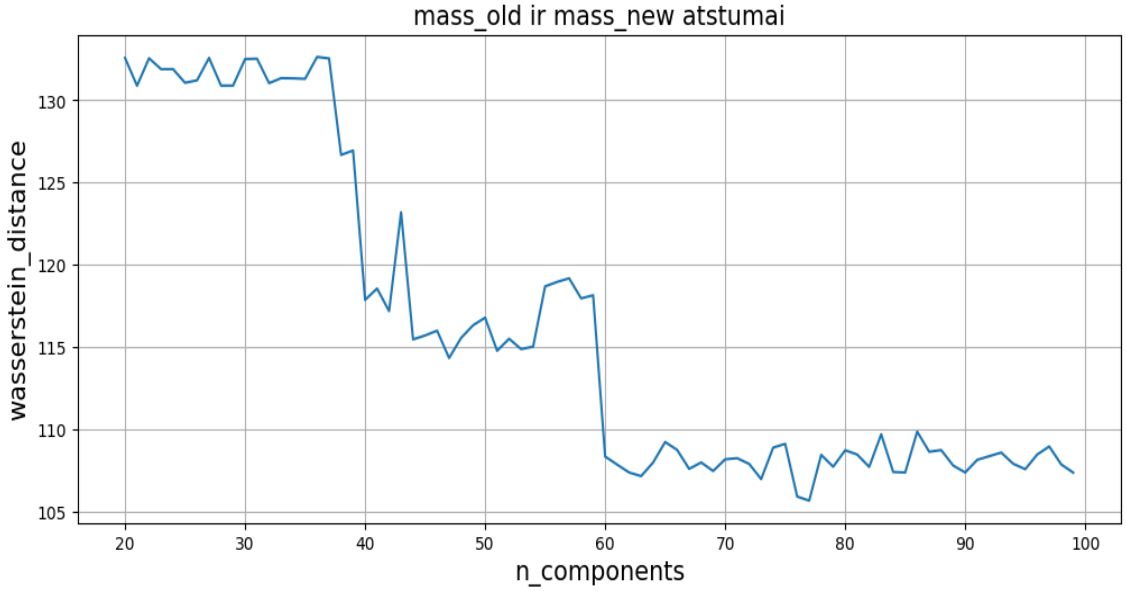


28 pav. Invariantinės masės m_{old} ir m_{new} palyginimas, kai yra naudojama testinė X_{test} aibė

Diagramoje 28 pav. gaunasi panašus invariantinės masės rezultatas, kaip diagramoje 27 pav. Yra didelių masių skirtumų. Kadangi X_{test} reikšmės sudaro 20% *Monte Carlo* sugeneruotų duomenų rinkinio, $mass_{old}$ ir $mass_{train}$ aibių ilgiai buvo suvienodinti ir lygūs testinės aibės X_{test} ilgiui. Invariantinė masės aibę $mass_{new}$, kuri yra vaizduojama grafike 28 pav., turi klaidingų reikšmių 1241 $mass_{new}$ aibėje iš esančių 4000 reikšmių. Norint, *NaN* reikšmių neatvaizduoti masių aibių palyginimų histogramose, $mass_{new}$ aibėje *NaN* vertės buvo pakeičiamos į 0. Toks principas pritaikytas tam, kad būtų aiškesni aibės $mass_{new}$ rezultatai.

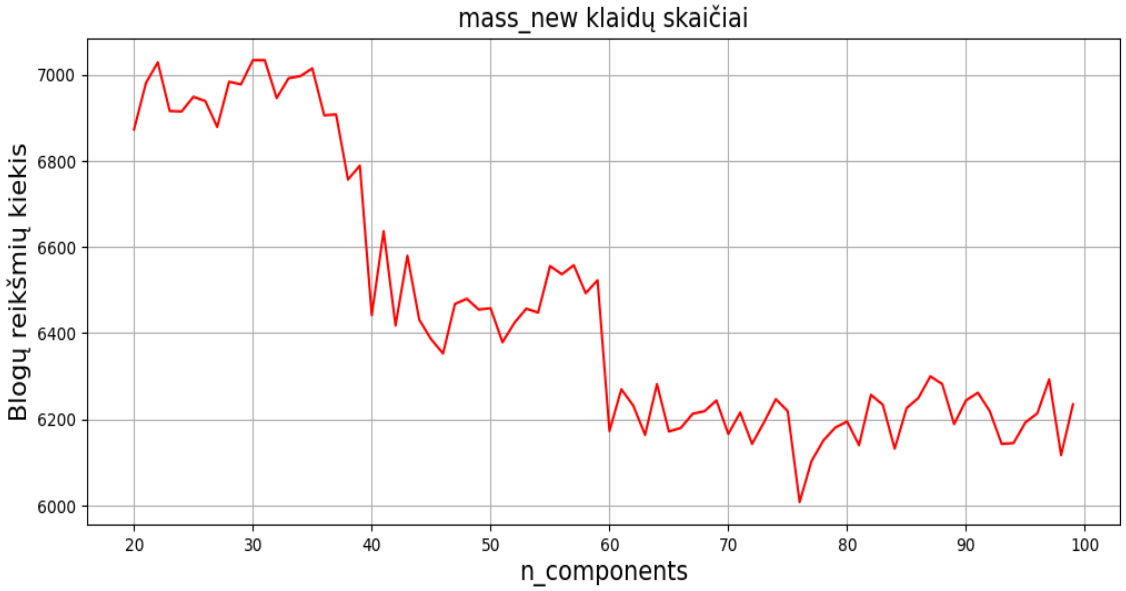
4.5.1. *Gauso* maišos modelio duomenų generavimo rezultatai

Atlikus hiperparametrų ir duomenų generavimo eksperimentus, pavyko atrasti hiperparametrų poras *n_components* ir *covariance_type*, kurios greitai sugeneruoja naują dalelių duomenų rinkinį, bet su mažesniu tikslumu. Tačiau, yra įdomu, kokios dar gali būti tinkamos *n_components* reikšmės *GMM* modeliui?



29 pav. Invariantinės masės $mass_{old}$ ir $mass_{new}$ atstumai pagal *wasserstein* atstumą

Grafikas 29 pav., rodo, kad tinkamiausi *n_components* būtų, kai *n_components* yra intervale $(75, 78)$. Linijinis grafikas rodo, kad šiame *n_components* intervale, *wasserstein* atstumas yra arti grafiko minimumo. Reiškia, kad taikant šias *n_components* vertes, pagal modelio *GMM* sugeneruotą duomenų rinkinį, turėtų gautis daugiau geresnė apskaičiuota $mass_{new}$ reikšmė ir būtų galima teigti, kad modelis geriau atlieka darbą.



30 pav. m_{new} aibės klaidingų reikšmių kiekiai

Žvelgiant į sugeneruotų invariantinės masės klaidingų reikšmių kiekio grafiką 30 pav. pagal skirtingą *n_components* vertę, padidėjus *n_components* reikšmei, pradeda mažėti prastų sugeneruotų reikšmių aibėje $mass_{new}$ kiekis. *n_components* intervale $(76, 80)$ klaidų kiekio grafikas pasiekia mažiausią reikšmę. Galima teigti, kad šio intervalo *n_components* reikšmės leistų *GMM* modeliui suteikti geresnių rezultatų.

Analizuojant grafikus 29 pav., 27 pav., 28 pav. ir 30 pav., galima teigti, kad *GMM* duomenų generavimo eksperimentai yra arti tinkamo tikslumo. Klaidingų reikšmių atsiradimas $mass_{new}$ aibėje lemia *GMM* modelio hyperparametrai. Jeigu pavyksta parinkti gerus hyperparametrus, pagerėja $mass_{new}$ skaičiavimo rezultatai. Norint pagerinti duomenų generavimo rezultatus, ateities planuose yra įtraukta išbandyti kitas kovariacijos matricos formas, kitokių hyperparametrų kombinacijų su *GMM* modelio *n_components* verte bei išsamiai pažiūrėti, kaip kinta klaidingų reikšmių kiekis $mass_{new}$ aibėje pagal *spherical*, *diag* ir *full* kovariacijos matricos formas.

4.5.2. *GMM* palyginimas su gilaus mokymo modelio rezultatais

Vienas iš geriausių duomenų generavimo eksperimentų rezultatų būtų normalizavimo srautų (*NF*) modelis. Modelio apmokymas trunka 4 minutes ir 30 sekundžių. Modelio invariantinės masės grafikas 12 pav. ir koreliacijos diagramų palyginimas pagal frobenijaus normą 26 pav. teigia, kad *NF* modeliui pavyksta beveik tiksliai sugeneruoti naujus fizikinius aukštos energijos dalelių duomenis, kurie turėtų fizikinių ir statistinių savybių, kurias galima pastebėti *Monte Carlo* būdu sugeneruotuose duomenyse. Dar ir gali patvirtinti *NF* modelio apskaičiuotas *FID* (*Frechet Inception distance*) atstumas pagal duotus *Monte Carlo* ir *VAE* būdu modeliu sugeneruotus duomenis. *VAE* modelio *FID* atstumas yra lygus $2,94186364465 * 10^9$.

Analizuojant *NF* netekties funkcijų grafikus (22 pav.) ir *GMM* modelio W_1 grafiką (29 pav.), matosi, kad modeliams pavyksta sugeneruoti tiksliau naujus fizikinius duomenis, kai turi tam tikras, dideles hiperparametrų parametrų reikšmes. *GMM* modeliui pavyksta tiksliau sugeneruoti naujus fizikinius duomenis, kai turi didesnę *n_components* reikšmę, o *NF* modeliui pavyksta geresnius, naujus fizikinius duomenis sugeneruoti, kai jo apmokymas trunka apie 120 epochų ir mokymosi dažnis η yra labai mažas, lygus 10^{-6}.

Pavyko nustatyti, kad nors ir paprasta *GMM* modelį realizuoti, apmokyti bei sugeneruoti, naujus fizikinius duomenis per trumpą laiką, trunkantį 6 sekundes, *NF* modeliui vis tiek pavyko sugeneruoti geriau naujus duomenis negu *GMM* modeliui. *GMM* modelis nereikalauja specialios architektūros. Reikia tik parinkti tinkamas hiperparametrų reikšmes susijusias su *n_components* ir *covariance_type*, žyminčia kovariacijos tipą pagal kurį modelis bandytų kurti naujų duomenų aproksimacijas. *NF* modelio apmokymo laikas didesnis už *GMM* modelio, bet *NF* modelis naudoja mažiau virtualios atminties ir sugeneruoja tiksliau *HEP* duomenis. *NF* modelio invariantinės masės atstumas yra mažesnis už *GMM* modelio. *GMM* modelio apmokymui reikia 245,80 megabaitų. *NF* modelio apmokymui užtenka *GPU* virtualios atminties, kuri yra lygi 172.032 megabaitai. *NF* modelis, kaip ir visi kiti gilaus mokymo modeliai, gali naudoti ir *GPU* vaizdo plokštę ir kompiuterio procesorių *CPU*. Yra galimybių naujų duomenų generavimo operacijas vykdyti *GPU* vaizdo plokštėje ir *CPU* procesoriuje. *GPU* plokštes skaičiavimai yra greitesni negu kompiuterio procesoriuje. Parinkus tinkamas *NF* modelio ir optimizavimo *Adam* parametrus, galima pagerinti modelio veikimą generuojant naujus fizikinius dalelių duomenis. Atliekant eksperimentus nustatyta, kad mažėjant mokymosi dažniui *Adam* algoritme gerėja rezultatai, kai *NF* architektūroje yra srautų perceptrono struktūrose didesnis neuronų kiekis viename srauto sluoksnyje ir maži neuronų kiekiai keliose srauto sluoksniuose.

Išvados

- *VAE* modelis turėtų greičiau veikti dėl mažo neuronų kiekio ir galimybių neprarasti naujų duomenų generavimo tikslumo dėl *ELU* aktyvacijos funkcijos [16]. Normalizavimo srautai naudoja tik sigmoidės aktyvacijos funkciją ir *GAN* modelis taiko *LeakyReLU* aktyvacijos funkciją netiesiškumo bruožų sukūrimui naujuose duomenyse. Atlikus *VAE* duomenų generavimo eksperimentą, pastebėta, kad *KL* divergencijos ir vidutinė kvadratinė paklaidos bei M_{inv} netekties funkcijos logaritmiškai mažėja kiekvienoje modelio mokymo ciklo epochoje. Patikrinus *VAE* invariantinės masės ir dalelių greičių p_T histogramas, pastebėta, kad *VAE* modeliui pavyksta sugeneruoti fizikinių dalelių energijas, dalelių įvykių z koordinates, bet ne įvykių x ir y koordinates. *el1_neg_px*, Naujame duomenų rinkinyje *el1_neg_py* ir kituose x ir y dalelių koordinačių stulpeliuose susidaro labai mažos, artėjančio iki 0 reikšmės (pavyzdžiui $0,6$, 0.5992 ir pan.). Ištyrus visų naudotų gilaus mokymo modelių rezultatus, gavosi normalizavimo srauto modelio geresni rezultatai. Įrodo panaudotos statistinės ir fizikinės metrikos.

- Normalizavimo srautų modeliui pavyko sugeneruoti geriau naujus fizikinius duomenis negu *VAE* ir *GAN* modeliui. Normalizavimo srautų eksperimentuose yra naudotas autoregresyvus transformacijos metodas modelio linijinės transformacijos sluoksniams. Suteikė modeliui galimybių tiksliau apskaičiuoti duotų fizikinių duomenų tikimybines, aproksimacijų vertes iš kurių galėjo sugeneruoti tinkamus naujus, fizikinių dalelių įvykių koordinačių x, y ir z bei energijų duomenis. Nesigavo labai mažos x ir y dalelių įvykių koordinatės kaip *VAE* modelyje. Sugeneruotuose normalizavimo srautų duomenyse yra didesnis standartinis nuokrypis negu *VAE* modeliu sugeneruotuose duomenyse. Pagal naujus fizikinius dalelių duomenis, kurie yra sugeneruoti pagal normalizavimo srautų apsiskaičiuoja tiksliau dalelių greičių reikšmės p_T, nes nėra labai mažas standartinis nuokrypis kaip *VAE* modelio dalelių greičių rezultatuose. Normalizavimo srautų modelio frobenijaus normos reikšmė rodo, jog sugeneruotų duomenų koreliacijos savybės yra arti duotų duomenų rinkinio, o *GAN* modeliu ir *VAE* modelių sugeneruotų duomenų frobenijaus normos reikšmės rodo, kad duomenų koreliacijos savybės nėra panašios į duotų, *Monte Carlo* metodu sugeneruotų duomenų.

- Apmokant *VAE*, *GAN* ir normalizavimo srauto modelius, pastebėta, kad pavyksta pagerinti modelių veikimo greitaveiką, kai mokymo bei naujų duomenų generavimo procesai vyksta grafinėje kortoje ir kompiuterio procesoriuje. Modeliai buvo paleisti *Google Colab* ir universiteto superkompiuteryje. Internetinėje *Google Colab* sėkmingai pavyko lygiagrečiai vykdyti duomenų krovimo, modelio treniravimo skaičiavimus ir pan naudojant paslaugos teikiamą grafinę kortą *NVIDIA T4 Tensor core* ir kompiuterio procesorių *Intel(R) Xeon(R) 2,20 Ghz* greičio. Dėl grafinės kortos naudojimo pagreitėjo modelio apmokymo ir naujų duomenų generavimo skaičiavimų rezultatai. Universiteto superkompiuteryje buvo naudota aplinka, kurioje buvo prieinamas tik procesorius 8 branduolių, *Intel(R) Xeon(R) 2,10 Ghz* greičio. Kai kurių modelio naujų duomenų generavimo skaičiavimai vyko lėčiau, nes procesoriuje vyksta sinchroniniai skaičiavimai ir būna mažiau branduolių negu grafinėje kortoje. *VAE* modelio *Google Colab* ir superkompiuterio skirtumas yra 21 minutės ir 28 sekundės. *GAN* modelio skirtumas būtų 6 sekundės ir normalizuojančių srautų būtų 5 minutės ir 1 sekundė. Norint pagerinti pasirinktų modelių greitaveikos rezultatus, reikėtų parinkti geresnę modelių architektūrą, kurioje būtų mažiau sluoksnių ir neuronų, tinkamus optimizavimo algoritmų hiperparametrus (mokymosi dažnį, svorių mažėjimo koeficientą ir pan.) tam, kad būtų apskaičiuoti geresni linijinių transformacijų svoriai.

- Panaudota, modelių mokymo metu, gradientų reikšmių skaičiavimams invariantinės masės netekties funkcija, nepadėjo pagerinti naujų duomenų generavimo rezultatų. *VAE*, *GAN* ir normalizavimo srautų modelio sugeneruotuose naujuose, fizikiniuose, aukštos energijos dalelių duomenyse susidaro labai mažos reikšmės, kurios eksperimentų koduose yra apibrėžiamos kaip *NaN* (*Not a number*) vertės. Tokios vertės atsiranda, nes *Adam* algoritmo momentinių vektorių skaičiavimuose susidarė dalybą iš 0. Buvo patikrintas netekties funkcijos realizuotas kodas, nebuvo klaidų. Norint išspręsti šią problemą, turėtų būti geriau patobulinta modelio architektūrą ir hiperparametrai tam kad išeitų geriau panaudoti invariantinės netekties funkciją.

- Dirbant prie gilaus mokymo modelių veikimo analizės ir duomenų generavimo eksperimentų, sužinota, kad *GAN*, *VAE* ir normalizuojančių srautų modeliai turi tikslą transformuoti duotus duomenis į latentinės erdvės z reikšmes iš kurių būtų sugeneruojami nauji duomenys Y. Ištirta, kad kiekvienas modelis gali ne visada tiksliai sugeneruoti naujus fizikinius dalelių duomenis. *VAE* modeliui pavyksta sugeneruoti naujuose dalelių duomenyse fizikines dalelių energijas, z koordinates, o ne x ir y koordinates. *GAN* modelis gali tinkamai sugeneruoti naujus duomenis, jeigu būtų parinktą geresnė modelio architektūra. Normalizavimo srautų modelis, gali visus fizikinius dalelių duomenis sugeneruoti, tik kai kurios energijų reikšmės susidaro mažos, nes apsiskaičiuoja 0 invariantinės masės reikšmės. Išmokta, kad norint pagerinti modelių naujų duomenų generavimo rezultatus, reikia mažinti *Adam* algoritmo mokymosi dažnį η, taikyti linijines aktyvacijos funkcijas, kurios padėtų modeliui sukurti netiesiškumo bruožų naujuose duomenyse ir taikyti duomenų normalizavimo metodus modeliams, tam kad būtų greičiau apmokami pasirinkti modeliai.

Ateities tyrimų planas

- Pagal fizikines ir statistines metrikas, palyginti, sugeneruotus naujus, aukštos energijos fizikinius duomenis su realiais fizikiniais dalelių duomenimis, kurie yra gauti *CERN* dalelių *LHC* greitintuve.
- Išanalizuoti kaip veikia *Monte Carlo* generatoriai: *PYTHIA*, *HERWIG*, *COJETS* ir t.t. Nustatyti ar sugeneruoti nauji fizikiniai aukštos energijos duomenys pagal gilaus mokymo modelius yra geresni negu *Monte Carlo* generatorių. Įvertinti greitaveiką *Monte Carlo* generatorių ir gilaus mokymo modelių.
- Atlikti detalią modelių veikimo greitaveikos analizę. Sužinoti, kurios modelių funkcijos daugiausiai naudoja kompiuterio procesoriaus laiko naujų duomenų generavimo skaičiavimuose. Atradus šias funkcijas, pakeisti į kitas ir nustatyti iš naujo ar modelio greitaveiką pagerėja.
- Patobulinti taikytų modelių *VAE*, *GAN* ir normalizuojančių srautų modelių naujų duomenų generavimo architektūras. Atrasti modelių architektūros variantą, kuris padėtų dar tiksliau generuotų naujus fizikinius aukštos dalelių duomenis.
- Išbandyti daugiau hiperparametrų algoritmų, kurie padėtų patobulinti taikytų modelių naujų generavimui svorius. Palyginti jų veikimo principus ir nustatyti, kuris būtų labiau tinkamesnis modelių apmokymų ciklų rezultatus gerinti.

Literatūros šaltiniai

[1] Josh McFayden1 Simone Amoroso Joshua Bendavid Andy Buckley Matteo Cacciari Taylor Childers Vitaliano Ciulli Rikkert Frederix Stefano Frixione Francesco Giuli Alexander Grohsjean Christian Gütschow Stefan Höche Walter Hopkins Philip Ilten Dmitri Konstantinov Frank Krauss Qiang Li Leif Lönnblad Fabio Maltoni Michelangelo Mangano Zach Marshal Olivier Mattelaer Javier Fernandez Menendez Stephen Mrenna Servesh Muralidharan Tobias Neumann Simon Plätzer Stefan Prestel Stefan Roiser Marek Schönherr Holger Schulz Markus Schulz Elizabeth Sexton-Kennedy Frank Siegert Andrzej Siódmok Graeme A. Stewart Andrea Valassi, Efe Yazgan. Challenges in monte carlo event generator software for high-luminosity lhc. Computing and Software for Big Science (2021) 5:12, 2021. https://arxiv.org/abs/2004.13687.

[2] Isabell-A Melzer-Pellmann Moritz Scham Simon Schnake Benno Käch, Dirk Krücker and Alexi Verney-Provatas. Jetflow: Generating jets with conditioned and mass constrained normalising flows. Deutsches Elektronen-Synchrotron DESY, Germany, 2023. https://arxiv.org/abs/2211.13630.

[3] Christopher M. Bishop. Bishop pattern recognition and machine learning. Microsoft Research Ltd, 2006. https://www.microsoft.com/en-us/research/uploads/prod/2006/01/Bishop-Pattern-Recognition-and-Machine-Learning-2006.pdf.

[4] Joshua Isaacson Claudius Krause Holger Schulz Christina Gao, Stefan Höche. Event generation with normalizing flows. Published by the American Physical Society, 2020. https://arxiv.org/abs/2001.10028.

[5] Jimmy Lei Ba Diederik P. Kingma. Adam: A method for stochastic optimization. Published as a conference paper at ICLR 2015, 2023. https://arxiv.org/abs/1412.6980.

[6] Max Welling Diederik P. Kingma. An introduction to variational autoencoders. Diederik P. Kingma and Max Welling (2019), "An Introduction to Variational Autoencoders", Foundations and Trends in Machine Learning: Vol. xx, No. xx, pp 1–18. DOI: 10.1561/XXXXXXXXX, 2019. https://arxiv.org/abs/1906.02691.

[7] Max Welling Diederik P. Kingma. Auto-encoding variational bayes. Auto-Encoding Variational Bayes, 2022. https://arxiv.org/abs/1312.6114.

[8] Iain Murray George Papamakarios, Theo Pavlakou. Masked autoregressive flow for density estimation. Masked Autoregressive Flow for Density Estimation, 2018. https://arxiv.org/abs/1705.07057.

[9] Julian Avila Trent Hauck. Scit-learn cookbook second edition. Birmingham B3 2PB, UK, 2017. https://github.com/shahumar/Free-Machine-Learning-Books/blob/master/book/scikit-learn%20Cookbook%20-%20Second%20Edition.pdf.

[10] Shiho Kim Hyunbin Parka. Hardware accelerator systems for artificial intelligence and machine learning. Copyright 2021 Elsevier Inc, 2020.
https://www.sciencedirect.com/science/article/abs/pii/S0065245820300929.

[11] Mehdi Mirza Bing Xu David Warde Farley Sherjil Ozair Aaron Courville Yoshua Bengio Ian J. Goodfellow, Jean Pouget-Abadie†. Generative adversarial nets. Departement d'informatique et de recherche op erationnelle Universite de Montreal, 2023.
https://arxiv.org/abs/1406.2661.

[12] Sandro Wenzel Przemys law Rokita Jan Dubi´nski, Kamil Deja and Tomasz Trzcinski. Machine learning methods for simulating particle response in the zero degree calorimeter at the alice experiment, cern. Warsaw University of Technology, 2023.
https://arxiv.org/pdf/2306.13606.pdf.

[13] Cheng Soon ong Marc Peter Deisenroth, A. Aldo Faisal. Mathematics for machine learning. Cambridge University Press (2020)., 2012.
https://mml-book.github.io/book/mml-book.pdf.

[14] Martin Heusel Hubert Ramsauer Thomas Unterthiner Bernhard Nessler. Gans trained by a two time-scale update rule converge to a local nash equilibrium. LIT AI Lab Institute of Bioinformatics, Johannes Kepler University Linz A-4040 Linz, Austria, 2018.
https://arxiv.org/abs/1706.08500.

[15] Humberto Reyes-González University of Genoa. Learning likelihoods of lhc results with normalizing flows. ACAT 2022 Bari, Italy, 24/10/2022, 2022.
https://indico.cern.ch/event/1106990/contributions/4996227/attachments/2533422/4360116/NFlikelihoods.pdf.

[16] PyTorch. Elu. Copyright 2023, PyTorch Contributors, 2023.
https://pytorch.org/docs/stable/generated/torch.nn.ELU.html.

[17] PyTorch. Relu. Copyright 2023, PyTorch Contributors, 2023.
https://pytorch.org/docs/stable/generated/torch.nn.ReLU.html.

[18] Hao Su Breno Orzari Thiago Tomei Maurizio Pierini Mary Touranakou Jean-Roch Vlimant Raghav Kansal, Javier Duarte. Particle cloud generation with message passing generative adversarial networks. 35th Conference on Neural Information Processing Systems (NeurIPS 2021), 2022.
https://arxiv.org/abs/2106.11535.

[19] Sebastian Raschka. Stat 453: Introduction to deep learning and generative models. Sebastian Raschka STAT 453: Intro to Deep Learning, 2020.
https://sebastianraschka.com/pdf/lecture-notes/stat453ss21/L04_linalg-dl_slides.pdf.

[20] Sana Ketabchi Haghighat Serena Palazzo Riccardo Di Sipio, Michele Faucci Giannelli. Dijetgan: A generative-adversarial network approach for the simulation of qcd dijet events at the lhc. University of Toronto, Canada, 2019.
https://arxiv.org/abs/1903.02433.

[21] Vahid Mirjall Sebastian Raschka. Python machine learning third edition. Copyright 2019 Packt Publishing, 2019.
https://www.packtpub.com/product/python-machine-learning-third-edition/9781789955750.

[22] Luiz Scheinkman Subutai Ahmad. How can we be so dense? the benefits of using highly sparse representations. How Can We Be So Dense? The Benefits of Using Highly Sparse Representations, 2019.
https://arxiv.org/pdf/1903.11257.pdf.

[23] Wieske de Swart Melissa van Beekveld Luc Hendriks Caspar van Leeuwen-Damian Podareanu Roberto Ruiz de Austri Rob Verheyen Sydney Otten, Sascha Caron. Event generation and statistical sampling for physics with deep generative models and a density information buffer. NATURE COMMUNICATIONS, 2020.
https://www.nature.com/articles/s41467-021-22616-z.

[24] Wieske de Swart Melissa van Beekveld Luc Hendriks Caspar van Leeuwen-Damian Podareanu Roberto Ruiz de Austri Rob Verheyen Sydney Otten, Sascha Caron. Event generation and statistical sampling for physics with deep generative models and a density information buffer. NATURE COMMUNICATIONS, 2021.
https://www.nature.com/articles/s41467-021-22616-z.

[25] Xuejiao Yang Ming Du Xin Li Xin Feng, Youni Jiang. Computer vision algorithms and hardware implementations: A survey. Published by Elsevier B.V, 2019.
https://www.sciencedirect.com/science/article/pii/S0167926019301762.

[26] Ali Zaman. Effect of the jet production on pseudorapidity, transverse momentum and transverse mass distributions of charged particles produced in pp collisions at tevatron energy. Chinese Physics C, 2023.
https://iopscience.iop.org/article/10.1088/1674-1137/39/7/073001/pdf.

[27] Jesús Zambrano. Gaussian mixture model - method and application. ResearchGate, 2017.
https://www.researchgate.net/publication/321245699_Gaussian_Mixture_Model_-_method_and_application.

Įvykę susitikimai su darbo vadovu

Data	Pavadinimas	Veikla
2023-06-27	Darbo plano sudarymas ir jo aptarimas	Su darbo vadovu apžvelgėme magistrinio darbo temos reikalavimus. Išanalizavome mokslo tiriamojo darbo rezultatus bei pastabas, kurios buvo gautos mokslo tiriamojo darbo gynimo metu.
2023-09-11	Pirminių *VAE* modelio implementacijų bandymų ir darbo aprašo aptarimas	Su darbo vadovu, aptarėme pirmojo pasirinkto mašininio mokymo modelio *VAE* implementacijų metodų pasirinkimus, kurie yra susiję su *PyTorch* ir *TensorFlow* bibliotekomis. Pakalbėjome kokios pirminės pasirinktos metrikos modelių veikimo vertinimui yra tinkamos. Dar ir susitikimo metu, aptarėme kaip tinkamiau rengti darbo aprašą. Padiskutavome, kokias vietas reiktų iš mokslo tiriamojo darbo perkelti į magistro darbo aprašą. Dar ir su darbo vadovu pakalbėjome, kokias dalis reiktų pradėti aprašinėti magistro darbo apraše.
2023-09-25	Darbo aprašo ir pirminių *VAE* modelio eksperimentų analizavimas	Susitikimo metu, aptarėme įvykdytų *VAE* mašininio mokymo modelio eksperimentų rezultatus. Aptarėme, jog reikia patobulinti esamą implementuotą modelio architektūrą. Nusprendėme, kad reikia panaudoti daugiau sluoksnių modelio sudėtingumo didinimui bei panaudoti geresnes treniravimo vertinimo metrikas. Dar ir apžvelgėme pradinį darbo aprašo įvado variantą. Darbo vadovas pakomentavo, jog mintys naudojamos įvade yra tinkamos bei paminėjo, kokias vietas įvade, reikia tinkamiau aprašinėti.
2023-10-09	*VAE* modelio eksperimentų rezultatų ir darbo aprašo rengimo aptarimas	Su darbo vadovu susitikimo metu, apžvelgėme naują pasirinktą *VAE* modelio architektūros implementaciją duomenų generavimo eksperimentams bei pirminius rezultatus, kuriuos sugeneravo. Nustatėme, kad nauji, gauti duomenų generavimo rezultatai yra netinkami, bet pasirinkta *VAE* architektūros implementacija yra geresnė. Dar ir su darbo vadovu buvo aptartas darbo aprašo planas skirtas susijusių darbų analizei. Nustatėme, kokius straipsnius reiktų aprašyti skyriuje ir kuriuos pasirinktus straipsnius reikės naudoti teorinėje darbo dalyje.

2023-10-23	*VAE* ir *GAN* modelio eksperimentų rezultatų ir darbo aprašo rengimo aptarimas	Susitikimo metu, su darbo vadovu, apžvelgėme *VAE* ir *GAN* modelių eksperimentų rezultatus bei darbo aprašo pradinę teorinę dalį ir susijusių darbų analizę. Aptarėme vietas kurias reikėtų pataisyti *VAE* ir *GAN* modelio eksperimentuose tam kad būtų geresni duomenų generavimo rezultatai. Pasitarėme, kurias vietas reikės pakoreguoti susijusių darbų analizėje, tam kad būtų tinkamiau parašytas tekstas. Susiplanavome darbus iki kito darbo aptarimo susitikimo.
2023-11-06	*VAE*, *B-VAE* ir *Normalizing Flows* modelio eksperimentų rezultatų bei darbo aprašo rengimo aptarimas	Susitikimo metu, su darbo vadovu, apžvelgėme *VAE*, *B-VAE* ir *Normalizing Flows* modelių eksperimentų rezultatus bei darbo aprašo aprašytą *VAE* ir *GAN* modelių teorinę dalį. Aptarėme teorinės dalies vietas kurias reikės iki kito susitikimo pataisyti. Nustatėme, kad *VAE* modelio rezultatai yra pakankamai geri ir beveik arti siekiamo tikslo. Pasitarėme, kurio mašininio mokymo modelio eksperimentų rezultatus tobulinsime bei kokie kiti darbai bus įgyvendinami magistro darbo rengimo metu iki kito susitikimo su darbo vadovu.
2023-11-20	Aptarti mašininio mokymo modelių eksperimentai ir darbo aprašas	Su darbo vadovu peržiūrėjome atliktus *VAE* modelio duomenų generavimo eksperimentus. Nustatyta, pagal padarytus statistinius grafikus, jog *VAE* modelis beveik tiksliai generuoja fizikinius duomenis, tik *loss* funkcijos reikšmė netinkamai mažėja. Iškeltas tikslas patobulinti duomenų generavimo modelio rezultatus iki kito susitikimo su darbo vadovu. Dar ir pasitarta kokius dar tam tikrus patobulinimus padaryti darbo apraše ir atliekamuosiuose eksperimentuose. Nutartas darbo veiksmų planas likusiam darbo rengimo laikui iki darbo gynimo dienos.
2023-12-13	Atliktų duomenų generavimo eksperimentų aptarimas	Aprodyti nuveikti duomenų generavimo eksperimentų darbai. Pristatyta vadovui, kokius darbus pavyko padaryti ir kaip sekėsi. Pasitarta, kokią praktinę mokslo tiriamojo darbo informacinę atliktų eksperimentų medžiagą patalpinti magistro darbo apraše bei kokius eksperimentų palyginimus aprašyti.

2023-12-27	Darbo aprašo ir atliktų eksperimentų aptarimas	Su darbo vadovu patikrinta esama darbo aprašo struktūra ir turinys. Su darbo vadovu pasitarta, kokius dar likusius duomenų generavimo eksperimentus padaryti. Aptarti atliktų modelių fizikinių dalelių duomenų eksperimentų rezultatai. Pažiūrėta kokius reikia pataisymus padaryti eksperimentų koduose. Vadovas pateikė rekomendacijų dėl darbo aprašo. Paminėjo kokią trūkstamą informaciją apraše reikėtų patalpinti.
2024-01-04	Paskutinių pataisymui aptarimas	Su darbo vadovu aptarta, kokius finalinius pataisymus reikia padaryti darbo apraše ir ką patikrinti bei pataisyti darytų *VAE*, *GAN* ir normalizuojančių srauto modelių eksperimentų koduose. Vadovas papasakojo kokios yra padarytos klaidos darbo apraše ir ką reikėtų jame pataisyti bei papildyti.