Modele Dyfuzyjne i Analiza Gęstości Probabilistycznej

Georgii Stanishevskii (w ramach projektu ID.UJ)
Opiekun Projektu Dr hab. Przemysław Spurek, prof. UJ

30 maja 2025

Streszczenie

Niniejszy projekt poświęcony jest badaniu eksperymentalnych metod estymacji logarytmu gęstości prawdopodobieństwa danych ($\log g(x)$) oraz związanej z nim funkcji score ($\nabla \log g(x)$) przy użyciu sieci neuronowych, w szczególności w kontekście modeli dyfuzyjnych. Analizowane są dwa główne podejścia: ”Metoda 1 - Dystylacja”, gdzie gradient wyjścia uczonej sieci U-Net stara się aproksymować wyjście zamrożonej, wytrenowanej sieci U-Net, oraz ”Metoda 2 - Bezpośrednie przyspieszone uczenie”, gdzie uczona jest sieć h_t, będąca kopią architektury oryginalnego U-Netu z ”Metody 1”, z niewielką skalarną głowicą konwolucyjną dodaną do sieci zamiast warstw Up-Blocks, w celu przyspieszenia procesu uczenia, tak aby gradient dla tak określonego wyjścia bezpośrednio aproksymował szum w standardowej funkcji kosztu modelu dyfuzyjnego. Przeprowadzono eksperymenty na zbiorze danych CIFAR-10, porównując uzyskane proxy dla logarytmu gęstości z innymi modelami generatywnymi, takimi jak Normalizujące Przepływy (Glow) oraz Modele Oparte O Energię (EBM), oraz z podejściem opartym na analizie odpowiedzi oryginalnej sieci U-Net na dane wejściowe z dodanym niewielkim szumem. Oceniano zdolność metod do detekcji próbek spoza rozkładu (OOD) oraz korelację oszacowań gęstości. Wstępne wyniki, w tym niska korelacja metryki V_0 z Metody 1 z modelami Glow oraz EBM, oraz wysoka wartość funkcji kosztu w etapach uczenia Metody 2, wskazują na złożoność zadania i sugerują obszary dla dalszych badań nad stabilnością i efektywnością proponowanych metod. Duża korelacja wyjścia wytrenowanego modelu z ”Metody 1” do metryki wykorzystującej wyjscie oryginalnego U-Netu po dodaniu niewielkiej ilości szumu również wymaga głębszej analizy oraz teoretycznego uzasadnienia.

Spis treści

Rozdział 1

Wprowadzenie

1.1 Cel i motywacja projektu

Głównym celem niniejszego projektu jest zbadanie i opracowanie metod estymacji logarytmu gęstości prawdopodobieństwa danych, oznaczanego jako $\log g(x)$, oraz związanej z nim funkcji score, $s(x) = \nabla \log g(x)$. Zrozumienie i umiejętność modelowania tych wielkości ma kluczowe znaczenie w kontekście generatywnych modeli dyfuzyjnych [1], [2], [3], [4], otwierając możliwości dla polepszonej generacji próbek, detekcji anomalii oraz potencjalnie nowych, szybszych algorytmów uczenia [5], [6]. Projekt koncentruje się na eksperymentalnym podejściu do uczenia sieci neuronowych, których wyjścia lub gradienty wyjść mogą służyć jako aproksymacje tych fundamentalnych wielkości.

1.2 Modele dyfuzyjne – krótki przegląd

Modele dyfuzyjne, takie jak DDPM (Denoising Diffusion Probabilistic Models) [1], stanowią klasę potężnych modeli generatywnych. Ich działanie opiera się na dwóch procesach:

- **Proces prosty (forward process):** Stopniowe dodawanie szumu Gaussowskiego do danych x_0 pochodzących z rzeczywistego rozkładu, aż do uzyskania czystego szumu x_T po T krokach. Dla danego x_0, zaszumiony obraz x_t na kroku t można opisać jako $x_t = \sqrt{\bar{\alpha}_t}x_0 + \sqrt{1-\bar{\alpha}_t}\epsilon$, gdzie $\epsilon \sim \mathcal{N}(0, I)$, a $\bar{\alpha}_t$ jest parametrem harmonogramu zaszumiania [1].
- **Proces odwrotny (reverse process):** Uczenie sieci neuronowej (typowo o architekturze U-Net, oznaczanej jako $\epsilon_\theta(x_t, t)$ lub $\phi(t, x_t)$) przewidywania i usuwania szumu na każdym kroku t, aby z x_t zrekonstruować x_{t-1}, ostatecznie generując próbkę x_0 z szumu x_T [1], [7].

1.3 Podstawy teoretyczne

Kluczową ideą teoretyczną, badaną w tym projekcie, jest związek między wyjściem sieci U-Net $\phi(t, x_t)$ (która w standardowych DDPM przewiduje szum

ϵ lub jest z nim bezpośrednio związana) a funkcją score $s_t(x_t) = \nabla_{x_t} \log p_t(x_t)$ (gradientem logarytmu gęstości danych p_t na kroku t procesu dyfuzji). W sensie matematycznym okazuje się, że $\phi(t, x_t)$ jest proporcjonalne do $s_t(x_t)$ (często $\phi(t, x_t) \propto -s_t(x_t)$, jeśli ϕ przewiduje szum) [2], [8], [9]. Istnieje więc hipoteza, według której, jeśli wytrenujemy sieć U-Net $h_t : \mathbb{R}^N \to \mathbb{R}$ tak, aby jej gradient $\nabla_{x_t} h_t(x_t, t)$ aproksymował funkcję score (jak w standardowym podejściu do uczenia modeli dyfuzyjnych, z tą różnicą, że zamiast zwykłego wyjścia sieci U-Net, w funkcji kosztu występuje jej gradient), to jej bezpośrednie wyjście będzie aproksymowało logarytm gęstości:

$$\log p_t(x_t) \approx h_t(x_t, t) + C(t). \tag{1.1}$$

1.4 Badane metody

W ramach projektu badane są głównie dwa podejścia do uczenia takiej sieci h_t lub sieci z nią związanej:

1. **Metoda 1 (”Dystylacja”):** Uczenie sieci ”studenta” (`trainable_unet`) tak, aby pewna transformacja jej wyjścia (konkretnie, gradient sumy wszytstkich elementów tensora jej wyjścia przewidującego szum) była zgodna z wyjściem $\phi(t, x_t)$ zamrożonej sieci ”eksperta” (`frozen_unet`).
2. **Metoda 2 (”Bezpośrednie uczenie”):** Uczenie sieci h_t z głowicą skalarną tak, aby jej gradient $\nabla_{x_t} h_t(x_t, t)$ bezpośrednio aproksymował prawdziwy szum ϵ_{true} dodany w procesie dyfuzj.

Rozdział 2

”Metoda 1 – Dystylacja” dla U-Net

2.1 Cel i architektura

Celem tego eksperymentu, było wytrenowanie sieci `trainable_unet` (o architekturze U-Net, identycznej z modelem `google/ddpm-cifar10-32` [10]) w taki sposób, aby gradient jej wyjścia (gradient sumy wszytstkich elementów tensora jej wyjścia przewidującego szum) względem zaszumionego wejścia x_t aproksymował bezpośrednie wyjście (predykcję szumu) zamrożonej, wstępnie nauczonej sieci `frozen_unet`.

2.2 Funkcja kosztu

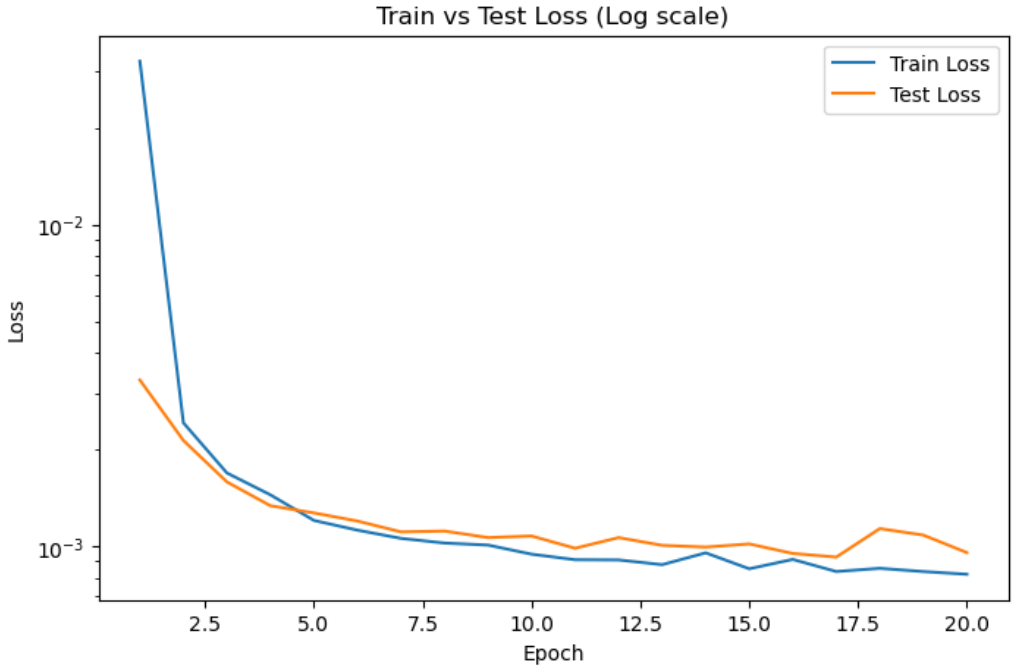
Funkcja kosztu, zdefiniowana jako `gradient_loss`, miała postać:

$$L(x_t, t) = ||\nabla_{x_t} \left(\sum \texttt{trainable\_unet\_output}(x_t, t) \right) - \texttt{frozen\_unet\_output}(x_t, t)||^2_{MSE} \quad (2.1)$$

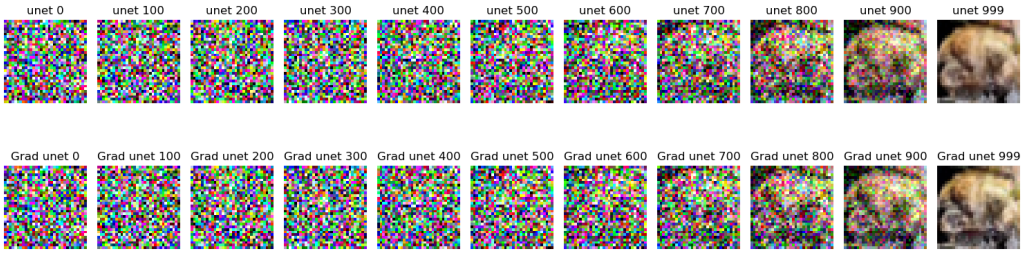
gdzie `trainable_unet_output` i `frozen_unet_output` to predykcje szumu (tensory o kształcie obrazu) odpowiednich sieci. W procesie uczenia optymalizowane były wszystkie parametry `trainable_unet`.

2.3 Proces uczenia

Wyuczenie modelu w 20 epokach zajęło ponad 24 godzin na NVIDIA RTX 4070M Funkcja kosztu dość szybko osiągała minimum, co sugeruje, że proces uczenia jest stabilny.



Rysunek 2.1: Wykres funkcji kosztu w czasie uczenia.



Rysunek 2.2: Porównanie wyników predykcji szumu dla modeli ”ekspert” (`frozen_unet, unet`) i ”student” (`trainable_unet, Grad unet`).

2.4 Weryfikacja hipotezy o estymacji gęstości (V_0)

Na podstawie analizy funkcji kosztu wysunięto hipotezę, że skalarna wielkość $V_0(x_0) = \sum \texttt{trainable\_unet}(x_0, t = 0)[0]$ (suma wszystkich komponentów tensora wyjściowego `trainable_unet` dla czystego obrazu x_0 przy $t = 0$) może być liniowo skorelowana z logarytmem gęstości $\log p_0(x_0)$, tj. $V_0(x_0) \approx -k \cdot \log p_0(x_0) + C(0)$.

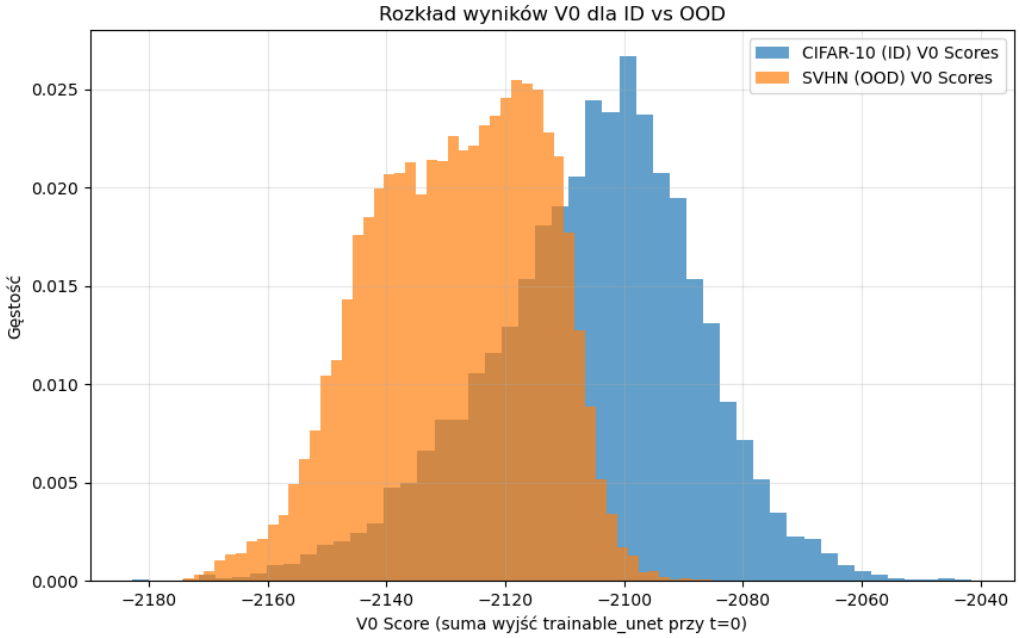
2.4.1 Eksperyment OOD dla V_0

Przeprowadzono eksperyment detekcji próbek spoza rozkładu (OOD), używając CIFAR-10 [11] jako danych ID oraz SVHN [12] jako danych OOD.

- Średnia wartość V_0 dla CIFAR-10 (ID): -2105.54 (Std: 17.66)

- Średnia wartość V_0 dla SVHN (OOD): -2128.32 (Std: 13.73)

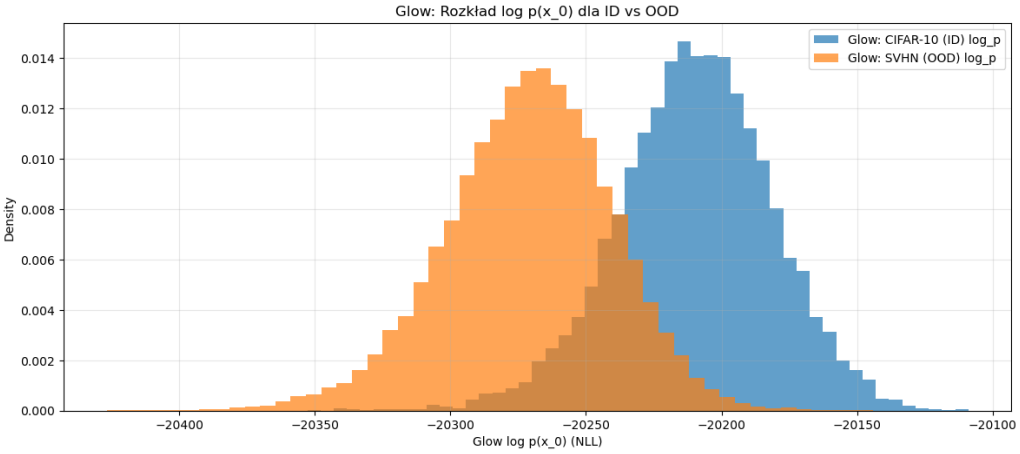
Wyniki te wskazują, że metryka V_0 jest w stanie odróżnić dane ID od OOD, co jest pozytywnym sygnałem na to, że uzyskana metryka może odzwierciedlać pewne aspekty ”typowości” danych



Rysunek 2.3: Rozkład wartości V_0 dla próbek ID (CIFAR-10) i OOD (SVHN).

2.4.2 Porównanie V_0 z modelem Glow

Model Glow (Normalizujący Przepływ wytrenowany na CIFAR-10) został użyty jako jeden z modeli referencyjnych dostarczający dokładnych wartości $\log p(x_0)$ [13].

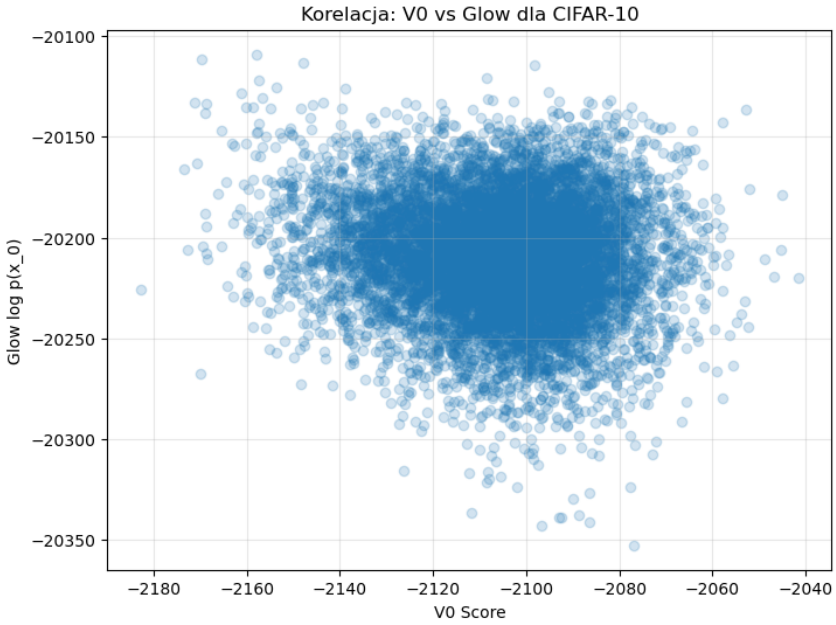


Rysunek 2.4: Rozkład wartości $\log p_{glow}$ dla próbek ID (CIFAR-10) i OOD (SVHN).

- Średnia $\log p_{glow}$ dla CIFAR-10 (ID): -20209.78 (Std: 28.75)

- Średnia $\log p_{glow}$ dla SVHN (OOD): -20271.20 (Std: 30.84)

Model Glow również efektywnie rozdziela dane ID i OOD.
Korelacja między V_0 a $\log p_{glow}$ na zbiorze CIFAR-10:



Rysunek 2.5: Wykres rozrzutu V_0 vs $\log p_{glow}$ na CIFAR-10.

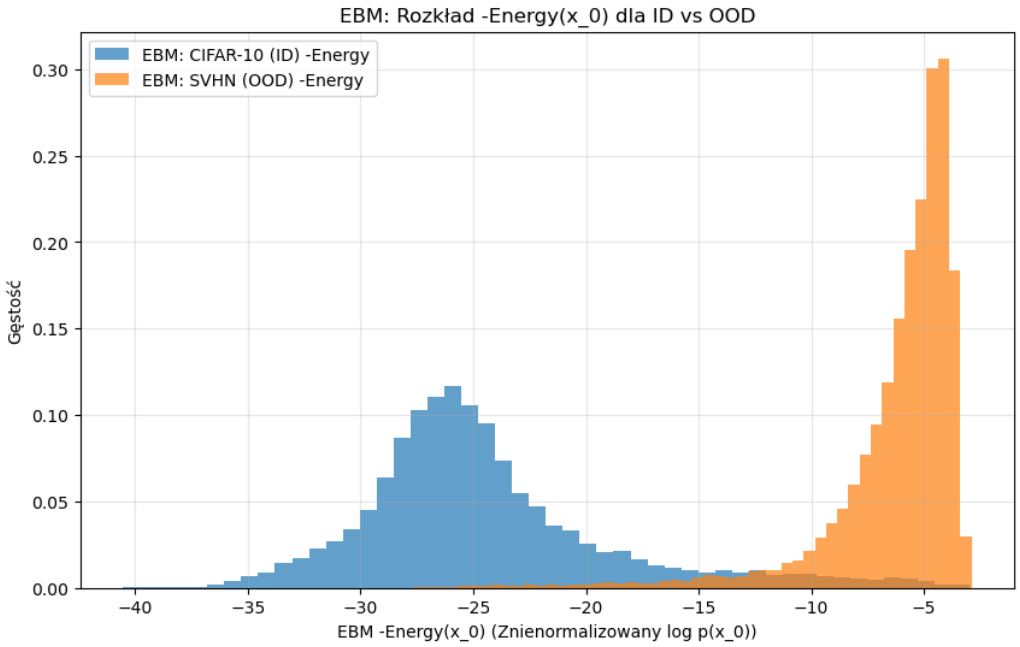
- Korelacja Pearsona: -0.0983
- Korelacja rangowa Spearmana: -0.0745

Niska korelacja sugeruje, że metryka V_0 (z Metody 1) i $\log p_{glow}$ inaczej oceniają ”typowość” obrazów z CIFAR-10.

2.4.3 Porównanie V_0 z modelem EBM

W celu dalszej walidacji metryki V_0 (pochodzącej z Metody 1), porównano ją z Modelem Opartym na Energii (EBM). Wykorzystano model EBM wytrenowany na CIFAR-10, którego wartości energii $E(x_0)$ (lub ich negacje) służą do oceny typowości danych [14]. Niższe wartości energii odpowiadają danym bardziej typowym (ID).

Wyniki detekcji OOD przy użyciu EBM przedstawiono na Rys. 2.6.

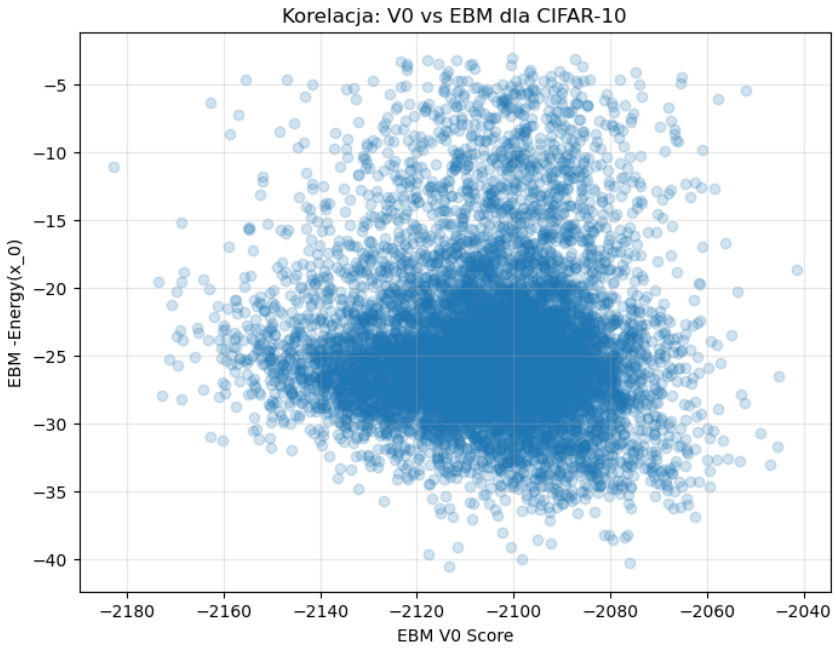


Rysunek 2.6: Rozkład wartości energii EBM dla próbek ID (CIFAR-10) i OOD (SVHN).

- Średnia wartość energii EBM dla CIFAR-10 (ID): -24.46 (Std: 5.64)
- Średnia wartość energii EBM dla SVHN (OOD): -5.64 (Std: 2.92)

Model EBM wyraźnie rozdziela dane ID i OOD.

Korelację między metryką V_0 a wartościami energii EBM na zbiorze CIFAR-10 ilustruje Rys. 2.8.

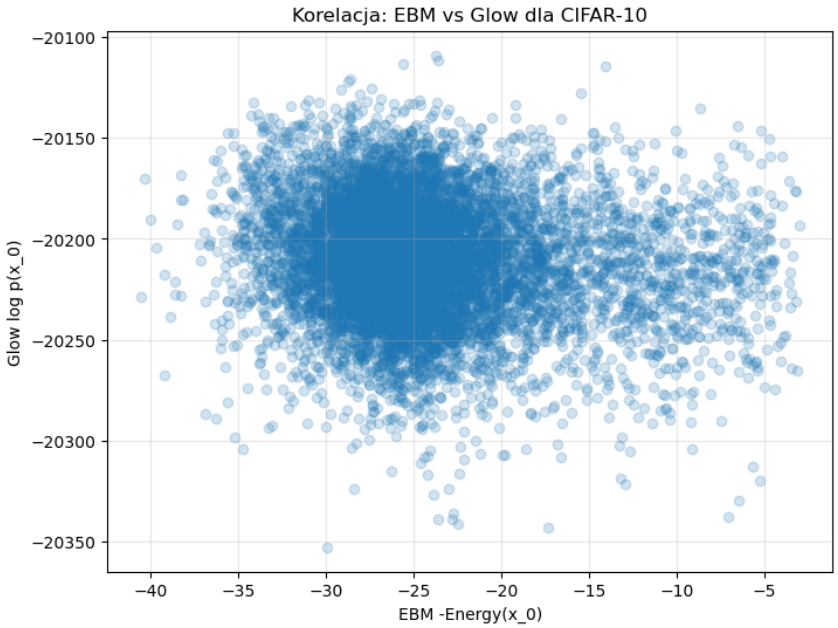


Rysunek 2.7: Wykres rozrzutu V_0 vs wartości energii EBM na CIFAR-10.

- Korelacja Pearsona: $-.0300$

- Korelacja rangowa Spearmana: -0.0405

Niska korelacja sugeruje, że metryka V_0 i energia EBM inaczej oceniają ”typowość” obrazów z CIFAR-10. Jednak to nie wyklucza hipotezy, że metryka V_0 może być pewną miarą typowości obrazów. Poniżej przedstawiono wykres korelacji pomiędzy modelami Glow i EBM na zbiorze CIFAR-10.



Rysunek 2.8: Wykres rozrzutu V_0 vs wartości energii EBM na CIFAR-10.

- Korelacja Pearsona: -0.1281
- Korelacja rangowa Spearmana: -0.1250

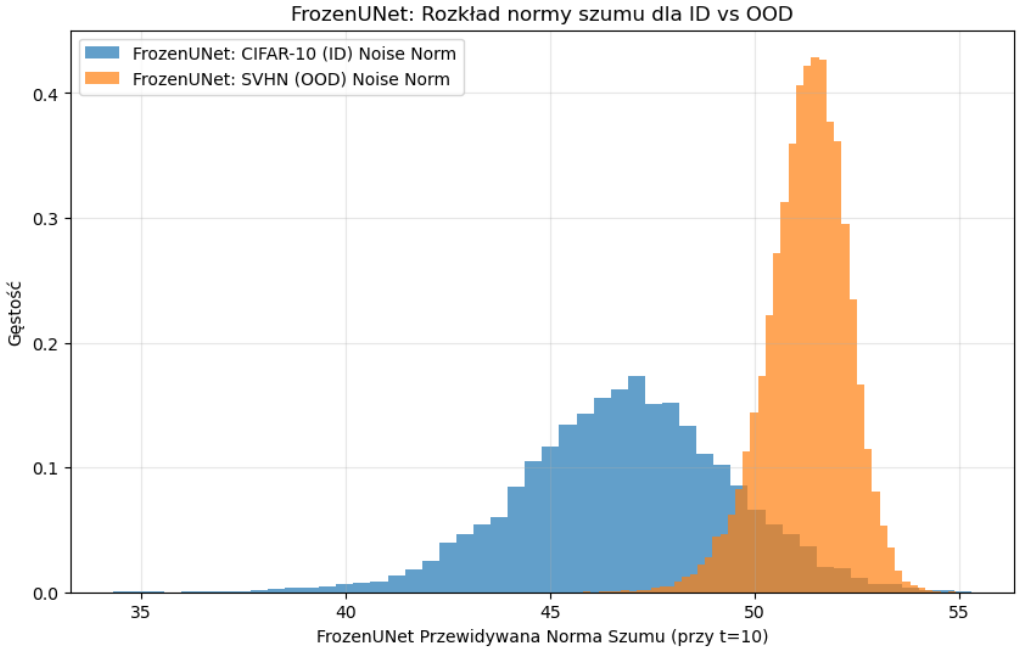
Jak widać z rysunku, chociaż korelacje pomiędzy Glow i EBM są nieco wyższe, niż w przypadku V_0, to jednak wciąż są one na niskim poziomie, co wskazuje na to, że oba modele referencyjne oceniają ”typowość” obrazów z innych perspektyw.

2.4.4 Porównanie V_0 z metryką ‘Noise Norm’

Jako trzecią metrykę referencyjną użyto normy L2 szumu przewidzianego przez `frozen_unet` (pretrenowany model ”ekspert” z ”Metody 1”) dla lekko zaszumionego obrazu x_0 (na kroku $t = 10$, t $\in \{1, 2, ..., 1000\}$),

$$S_{\text{frozen}}(x_0) \;=\; \left\| \texttt{frozen\_unet}\left(x_{t_{\text{small}}}, t_{\text{small}}\right) \right\|_2.$$

Tę samą ideę—mierzenie wielkości (lub normy) wektora szumu predyktora jako wskaźnika «typowości»—wykorzystywano m.in. w pracach [15], [16], [17].

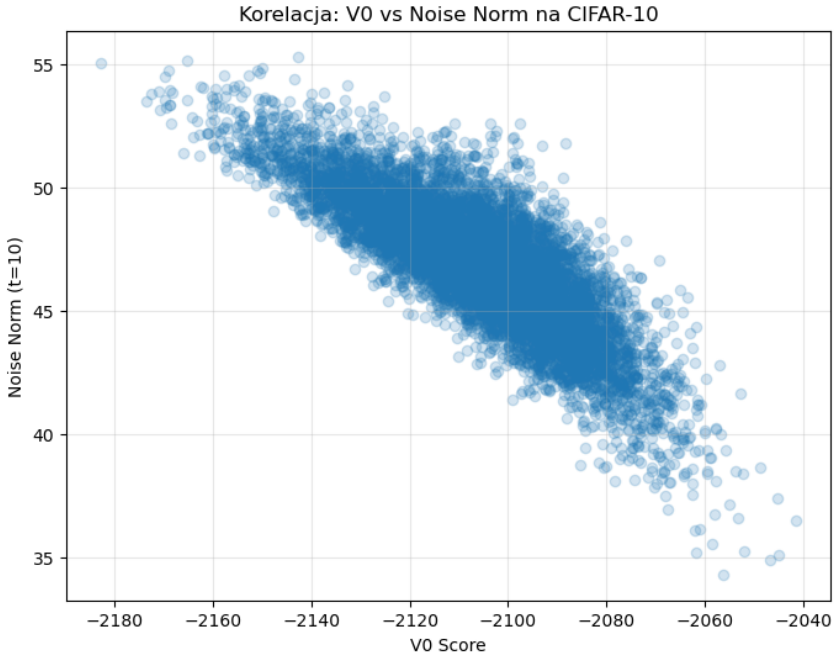


Rysunek 2.9: Rozkład wartości S_{frozen} dla próbek ID (CIFAR-10) i OOD (SVHN).

Wyniki detekcji OOD przy użyciu S_{frozen} przedstawiono na Rys. 2.9.

- Średnia S_{frozen} dla CIFAR-10 (ID): 46.81 (Std: 2.59)
- Średnia S_{frozen} dla SVHN (OOD): 51.28 (Std: 0.99)

Jak widać, metryka S_{frozen} w podobny sposób rozdziela dane ID i OOD.



Rysunek 2.10: Wykres rozrzutu S_{frozen} vs V_0 na CIFAR-10.

Korelację między S_{frozen} a V_0 na zbiorze CIFAR-10 ilustruje Rys. 2.10.

- Korelacja Pearsona: -0.8251
- Korelacja rangowa Spearmana: -0.8086

Wysoka korelacja między S_{frozen} a V_0 wskazuje, że istnieje związek pomiędzy normą szumu predykowanego dla lekko zaszumionego obrazu przez `frozen_unet` a wartością V_0 (suma wszystkich komponentów tensora wyjściowego `trainable_unet` dla czystego obrazu x_0 przy $t = 0$).

2.4.5 Wnioski dla V_0 z Metody 1

Metryka V_0 uzyskana z `trainable_unet` (uczona "Metodą 1") wykazuje pewną zdolność do detekcji OOD. Jej niska korelacja z etablowaną miarą gęstości $\log p(x_0)$ (z modelu Glow) może wskazywać, że V_0 w tej formie nie jest precyzyjnym przybliżeniem logarytmu gęstości, albo, że odzwierciedla pewne aspekty "typowości" danych w inny sposób niż Glow.

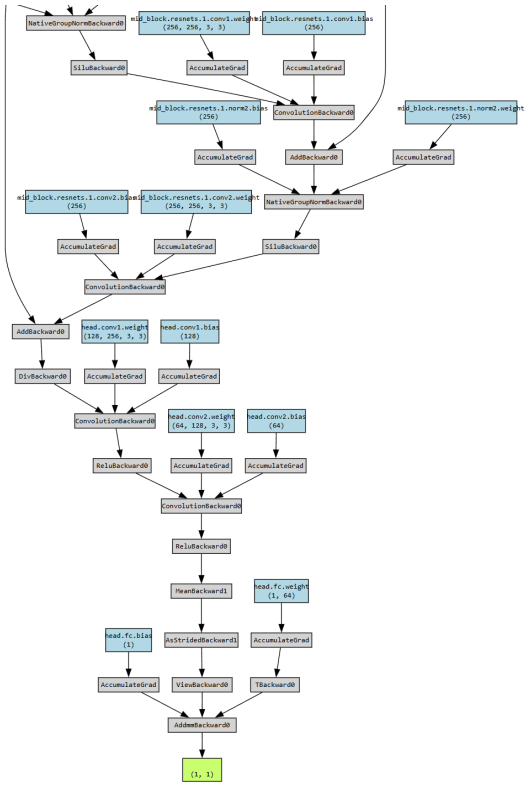
Rozdział 3

”Metoda 2 – Bezpośrednie uczenie gradientu uproszczonego modelu ze skalarną głowicą konwolucyjną”

3.1 Cel i architektura ScalarUNet

Celem ”Metody 2” jest bezpośrednie uczenie sieci h_t (nazwanej ‘ScalarU-Net‘), która generuje skalarny sygnał, tak aby jej gradient $\nabla_{x_t} h_t(x_t, t)$ mógł aproksymować rzeczywisty szum ϵ_{true} dodany w procesie dyfuzji. Architektura `ScalarUNet` składa się z:

- Ciała (Body): Zamrożone warstwy z `UNet2DModel` (`google/ddpm--cifar10-32`), obejmujące `time_proj`, `time_embedding`, `conv_in`, `down_blocks` oraz `mid_block`.
- Głowicy (Head): Uczona `ScalarHead`, która przyjmuje tensory cech z wyjścia `mid_block` c iała U-Net (liczba kanałów wejściowych dla głowicy 256). Głowica składa się z dwóch warstw konwolucyjnych z `BatchNorm` i `ReLU`, po których następuje `AdaptiveAvgPool2d(output_size=(1,1))`, `Flatten` oraz warstwa `Linear` dająca pojedynczy skalarny output.



Rysunek 3.1: Architektura `ScalarUNet` (ostatnie kilka warstw, całość modelu jest o wiele większa).

3.2 Funkcja kosztu

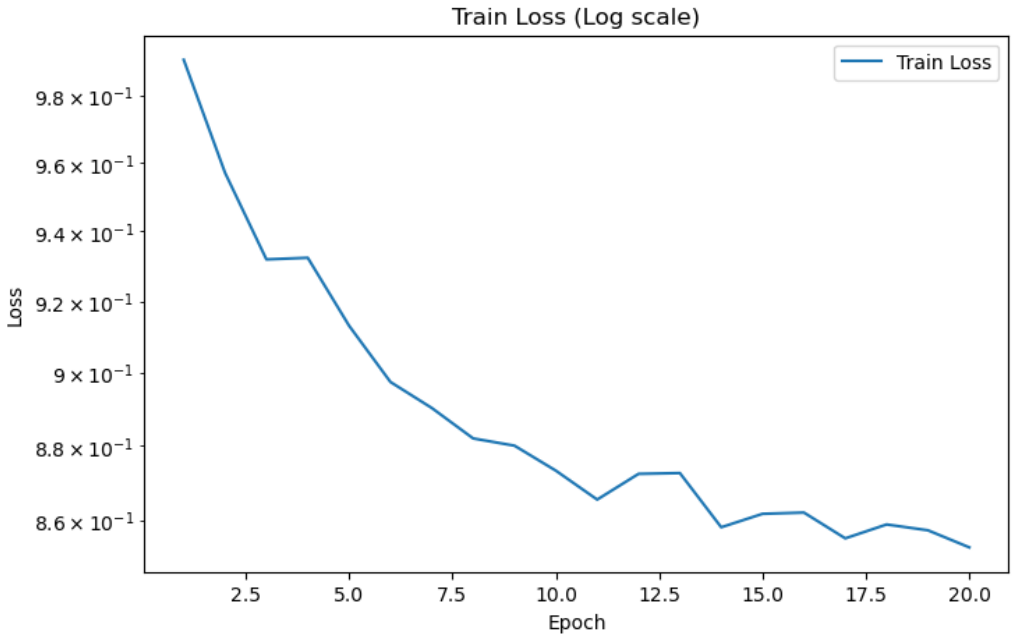
Funkcja kosztu dla ”Metody 2” to:

$$L(x_0, t, \epsilon_{true}) = ||\nabla_{x_t} h_t(x_t, t) - \epsilon_{true}||^2_{MSE} \tag{3.1}$$

gdzie $x_t = \sqrt{\bar{\alpha}_t} x_0 + \sqrt{1 - \bar{\alpha}_t}\epsilon_{true}$.

3.3 Proces uczenia (tylko głowica) i wyniki

Przeprowadzono uczenie modelu `h_t_model` (`ScalarUNet`) przez 20 epok, optymalizując jedynie parametry `ScalarHead` przy zamrożonym ciele początkowej części wytrenowanego U-Net. Użyto datasetu CIFAR-10, `batch_size=32`, `learning_rate=1e-4`. Wyniki uczenia przedstawiono na Rys. 3.2.



Rysunek 3.2: Wykres funkcji kosztu w zależności od epoki/kroku dla `ScalarUNet`.

Chociaż proces uczenia jest dużo szybszy (ok. 30 minut przeciwko ponad doby w ”Metodzie 1”), wartość funkcji kosztu pozostała stosunkowo wysoka (> 0.85) w porównaniu do wartości uzyskiwanych w ”Metodzie 1” (gdzie błąd spadał do ~ 0.001). Jest to oczekiwane, ponieważ celem jest aproksymacja ”surowego” szumu Gaussowskiego ϵ_{true} (o wariancji komponentów równej 1) za pomocą gradientu funkcji skalarnej, optymalizując jedynie głowicę oraz pozbywając się ciała początkowej części wytrenowanego U-Net, co jest trudnym zadaniem. Obserwowane zmniejszenie błędu, choć niewielkie, wskazuje, że proces uczenia głowicy postępuje. Należy rozważyć modyfikację architektury, np. pozwolenie na uczenie części wytrenowanego U-Net, bądź przywrócenie poprzedniej albo rozbudowę obecnej architektury `ScalarUNet`. Wtedy jednak możemy uzyskać mniejsze przyspieszenie względem Metody 1.

3.4 Dalsze kroki i analiza

- Kontynuacja uczenia `ScalarHead` na większą liczbę epok.
- Eksperymenty z architekturą `ScalarHead` oraz współczynnikiem uczenia.
- Po wstępnym nauczeniu głowicy, rozważenie rozmrożenia ciała U-Net i kontynuacja uczenia całej sieci `ScalarUNet` (z mniejszym współczynnikiem uczenia).
- Przeprowadzenie eksperymentu OOD oraz analizy korelacji dla bezpośredniego skalarnego wyjścia $h_t(x_0, t = 0)$ analogicznie do analizy V_0 z Metody 1.

Rozdział 4

Dyskusja Ogólna i Wnioski

W ramach projektu zbadano dwa podejścia do modelowania gęstości danych za pomocą modeli dyfuzyjnych.

Metoda 1, oparta na dystylacji gradientu w celu uzyskania metryki V_0, wykazała zdolność do detekcji OOD. Chociaż zaobserwowano niską korelację między V_0 a innymi modelami referencyjnymi, co sugeruje, że V_0 nie jest ich bezpośrednim przybliżeniem, V_0 może inaczej odzwierciedlać pewne aspekty "typowości" danych lub związek pomiędzy użytymi miarami gęstości może być bardziej złożony (metryki użytych modeli również nie wykazywały silnej korelacji między sobą). Jednakże, zaobserwowano silną korelację V_0 z normą szumu S_{frozen}, co wymaga dalszych badań oraz ścisłego wyprowadzenia matematycznego.

Metoda 2 polegała na bezpośrednim uczeniu sieci h_t (`ScalarUNet`) generującej sygnał skalarny, tak aby jej gradient aproksymował szum. Uczenie jedynie głowicy (`ScalarHead`) było znacznie szybsze niż w Metodzie 1, lecz funkcja kosztu pozostała wysoka. Wskazuje to na trudność zadania, ale również na postępujący proces uczenia. Pełna ocena tej metody wymaga dalszych analiz, w tym badania wyjścia $h_t(x_0, t = 0)$ oraz potencjalnej generacji obrazów.

Głównym celem projektu było uzyskanie sieci h_t aproksymującej $\log p_t(x_t)$. Metoda 1 nie osiągnęła tego celu bezpośrednio w kontekście korelacji z Glow, ale dostarczyła użytecznej metryki OOD. Metoda 2 jest bardziej bezpośrednim podejściem, ale jej skuteczność wymaga dalszej weryfikacji i potencjalnych modyfikacji, takich jak douczenie korpusu U-Net.

Dalsze badania powinny koncentrować się na głębszej analizie V_0, modyfikacjach architektury i procesu uczenia w Metodzie 2 (w tym rozważenie rozbudowy albo douczenia większej części sieci `ScalarUNet`), oraz na kompleksowej ocenie $h_t(x_0, t = 0)$ jako estymatora log-gęstości.

Bibliografia

[1] J. Ho, A. Jain i P. Abbeel, "Denoising Diffusion Probabilistic Models", w *Advances in Neural Information Processing Systems 33*, 2020, s. 6840–6851. DOI: 10.48550/arXiv.2006.11239. arXiv: 2006.11239. adr.: https://proceedings.neurips.cc/paper/2020/file/4c5bcfec8584af0d967f1Paper.pdf.

[2] Y. Song, J. Sohl-Dickstein, D. P. Kingma, A. Kumar, B. Poole i S. Ermon, "Score-Based Generative Modeling through Stochastic Differential Equations", w *International Conference on Learning Representations*, 2021. DOI: 10.48550/arXiv.2011.13456. arXiv: 2011.13456. adr.: https://openreview.net/forum?id=PxTIG12RRHS.

[3] P. Dhariwal i A. Nichol, "Diffusion Models Beat GANs on Image Synthesis", w *Advances in Neural Information Processing Systems 34*, 2021, s. 8780–8794. adr.: https://papers.nips.cc/paper_files/paper/2021/file/49ad23d1ec9fa4bd8d77d02681df5cfa-Paper.pdf.

[4] R. Rombach, A. Blattmann, D. Lorenz, P. Esser i B. Ommer, "High--Resolution Image Synthesis with Latent Diffusion Models", w *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, s. 10 684–10 695. DOI: 10.48550/arXiv.2112.10752. arXiv: 2112.10752. adr.: https://openaccess.thecvf.com/content/CVPR2022/papers/Rombach_High-Resolution_Image_Synthesis_With_Latent_Diffusion_Models_CVPR_2022_paper.pdf.

[5] Y. Song, C. Meng i S. Ermon, "Consistency Models", w *Proceedings of Machine Learning Research*, 2023, s. 11 108–11 140. DOI: 10.48550/arXiv.2303.01469. arXiv: 2303.01469. adr.: https://proceedings.mlr.press/v202/song23a/song23a.pdf.

[6] T. Salimans i J. Ho, "Progressive Distillation for Fast Sampling of Diffusion Models", w *International Conference on Learning Representations*, 2022. DOI: 10.48550/arXiv.2202.00512. arXiv: 2202.00512. adr.: https://openreview.net/pdf?id=TIdIXIpzhoI.

[7] J. Ho i T. Salimans, *Classifier-Free Diffusion Guidance*, 2022. DOI: 10.48550/arXiv.2207.12598. arXiv: 2207.12598. adr.: https://arxiv.org/abs/2207.12598.

[8] Y. Song i S. Ermon, "Improved Techniques for Training Score-Based Generative Models", w *Advances in Neural Information Processing Systems 33*, 2020, s. 12 438–12 448. DOI: 10.48550/arXiv.2006.09011. arXiv: 2006.09011. adr.: https://proceedings.neurips.

cc/paper/2020/file/92c3b916311a5517d9290576e3ea37ad-Paper.pdf.

[9] A. Hyvärinen, "Estimation of Non-Normalized Statistical Models by Score Matching", *Journal of Machine Learning Research*, t. 6, s. 695–709, 2005. adr.: https://jmlr.org/papers/volume6/hyvarinen05a/hyvarinen05a.pdf.

[10] Google Research, *google/ddpm-cifar10-32: Pre-trained DDPM on CIFAR-10 (32×32)*, https://huggingface.co/google/ddpm-cifar10-32, Model card, Hugging Face Hub, 2022. term. wiz. 4 czer. 2025.

[11] A. Krizhevsky, "Learning Multiple Layers of Features from Tiny Images", University of Toronto, spraw. tech., 2009. adr.: https://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf.

[12] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu i A. Y. Ng, "Reading Digits in Natural Images with Unsupervised Feature Learning", Stanford University, spraw. tech., 2011. adr.: http://ufldl.stanford.edu/housenumbers/.

[13] D. P. Kingma i P. Dhariwal, "Glow: Generative Flow with Invertible 1x1 Convolutions", w *Advances in Neural Information Processing Systems 31*, 2018, s. 10 215–10 224. DOI: 10.48550/arXiv.1807.03039. arXiv: 1807.03039. adr.: https://papers.nips.cc/paper/8224-glow-generative-flow-with-invertible-1x1-convolutions.pdf.

[14] Y. Du i I. Mordatch, "Implicit Generation and Modeling with Energy-Based Models", w *Advances in Neural Information Processing Systems 32*, 2019, s. 8468–8479. DOI: 10.48550/arXiv.1903.08689. arXiv: 1903.08689. adr.: https://papers.nips.cc/paper/8619-implicit-generation-and-modeling-with-energy-based-models.pdf.

[15] J. Zhou, A. Zhou i H. Li, "NODI: Out-of-Distribution Detection with Noise from Diffusion", *arXiv preprint arXiv:2401.08689*, 2024. adr.: https://arxiv.org/abs/2401.08689.

[16] A. Heng, A. H. Thiery i H. Soh, "Out-of-Distribution Detection with a Single Unconditional Diffusion Model", *arXiv preprint arXiv:2405.11881*, 2024. adr.: https://arxiv.org/abs/2405.11881.

[17] J. Wyatt, A. Leach, S. M. Schmon i C. G. Willcocks, "AnoDDPM: Anomaly Detection with Denoising Diffusion Probabilistic Models Using Simplex Noise", w *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2022, s. 650–656. adr.: https://openaccess.thecvf.com/content/CVPR2022W/NTIRE/papers/Wyatt_AnoDDPM_Anomaly_Detection_With_Denoising_Diffusion_Probabilistic_Models_Using_Simplex_CVPRW_2022_paper.pdf.