VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV TELEKOMUNIKACÍ

DEPARTMENT OF TELECOMMUNICATIONS

DIFUZNÍ NEURONOVÉ SÍTĚ PRO OPRAVU POŠKOZENÝCH AUDIO SIGNÁLŮ

DIFFUSION MODEL NEURAL NETWORKS FOR RESTORATION OF DAMAGED AUDIO

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE — Bc. Věra Podestátová

AUTHOR

VEDOUCÍ PRÁCE — Ing. Michal Švento

SUPERVISOR

BRNO 2025

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

Bakalářská práce

bakalářský studijní program **Audio inženýrství**
specializace Zvuková produkce a nahrávání
Ústav telekomunikací

Studentka: Bc. Věra Podestátová ***ID:*** 247800
Ročník: 3 ***Akademický rok:*** 2024/25

NÁZEV TÉMATU:

Difuzní neuronové sítě pro opravu poškozených audio signálů

POKYNY PRO VYPRACOVÁNÍ:

Nastudujte principy, na kterých funguje difúzní model hlubokého učení, který našel uplatnění i v úloze doplnění chybějícího úseku audio signálu (inpaintingu). Implementujte řešení z literatury [1] a navrhněte modifikace, které by mohly řešení vylepšit, nebo využít v jiné rekonstrukční úloze jako například declipping nebo zvyšování bitové hloubky. Vyberte dataset, který budete používat pro porovnávání výstupů algoritmů. Kvalitu rekonstrukce zhodnoťte pomocí objektivních metrik.

DOPORUČENÁ LITERATURA:

[1] E. Moliner, J. Lehtinen and V. Välimäki, "Solving Audio Inverse Problems with a Diffusion Model," ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Rhodes Island, Greece, 2023, pp. 1-5. https://ieeexplore.ieee.org/document/10095637

[2] E. Moliner and V. Välimäki. "Diffusion-Based Audio Inpainting," Journal of Audio Engineering Society, 2023, https://arxiv.org/abs/2305.15266

Termín zadání: 10.2.2025 ***Termín odevzdání:*** 3.6.2025

Vedoucí práce: Ing. Michal Švento

doc. Ing. Jiří Schimmel, Ph.D.
předseda rady studijního programu

UPOZORNĚNÍ:

Autor bakalářské práce nesmí při vytváření bakalářské práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č.40/2009 Sb.

ABSTRAKT

Bakalářská práce se zabývá využitím difuzního aposteriorního vzorkování pro rekonstrukci poškozených audio signálů se zaměřením na declipping. V rámci této metody jsou z Gaussovského šumu generovány vzorky audio signálu pomocí difuzního U-net modelu neuronové sítě, využívajícího Constant-Q transformaci. Kvalita rekonstrukce je otestována objektivními metrikami. Výsledky práce jsou v závěru zhodnoceny.

KLÍČOVÁ SLOVA

Difuzní aposteriorní vzorkování, difuzní neuronové sítě, rekonstrukce audio signálu, audio declipping, inverzní úlohy, hluboké strojové učení

ABSTRACT

The bachelor thesis deals with the use of diffusion posterior sampling for the reconstruction of corrupted audio signals with a focus on declipping. With this method, audio samples are generated from Gaussian noise using a U-net model of a diffusion neural network employing the Constant-Q transform. The reconstruction quality is tested by objective metrics. The results of the work are finally evaluated.

KEYWORDS

Diffusion posterior sampling, diffusion neural networks, audio signal reconstruction, audio declipping, inverse problems, deep learning

Vysázeno pomocí balíčku `thesis` verze 4.11; `https://latex.fekt.vut.cz/`

PODESTÁTOVÁ, Věra. *Difuzní neuronové sítě pro opravu poškozených audio signálů*. Bakalářská práce. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací, 2025. Vedoucí práce: Ing. Michal Švento

Prohlášení autora o původnosti díla

Jméno a příjmení autora:	Bc. Věra Podestátová
VUT ID autora:	247800
Typ práce:	Bakalářská práce
Akademický rok:	2024/25
Téma závěrečné práce:	Difuzní neuronové sítě pro opravu poškozených audio signálů

Prohlašuji, že svou závěrečnou práci jsem vypracovala samostatně pod vedením vedoucí/ho závěrečné práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autorka uvedené závěrečné práce dále prohlašuji, že v souvislosti s vytvořením této závěrečné práce jsem neporušila autorská práva třetích osob, zejména jsem nezasáhla nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědoma následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

Brno ....

podpis autorky*

*Autor podepisuje pouze v tištěné verzi.

PODĚKOVÁNÍ

Ráda bych poděkovala vedoucímu bakalářské práce panu Ing. Michalu Šventovi za odborné vedení, konzultace, trpělivost a podnětné návrhy k práci.

Obsah

Úvod **9**

1 Neuronové sítě **10**
1.1 Tenzory 11
1.2 Vrstvy 12
1.3 Aktivační funkce 13
1.4 Gradient 13
1.5 Optimalizace 14
1.6 Ztrátová funkce 14

2 Difuzní neuronové sítě **16**
2.1 Podmíněné a nepodmíněné modely 19

3 Inverzní úlohy **20**
3.1 Zvětšní šířky pásma 20
3.2 Doplnění chybějící části signálu 20
3.3 Declipping 21
3.4 Difuzní aposteriorní vzorkování 22

4 Výsledky **24**
4.1 Programové řešení 24
4.1.1 Dataset 24
4.1.2 Úprava vstupních dat 24
4.1.3 Difuzní čas a šumový rozptyl 26
4.1.4 Deterministický sampler 26
4.2 Hodnocení kvality rekonstrukce 27
4.2.1 Poměr signálu ke zkreslení 27
4.2.2 Střední absolutní chyba 28
4.2.3 Střední kvadratická chyba 29
4.2.4 Porovnání výstupů použitých metrik 30

Závěr **32**

Literatura **33**

Seznam symbolů a zkratek **37**

5 Obsah elektronické přílohy **38**

Seznam obrázků

1.1 Schéma propojení vrstev neuronové sítě. 10
1.2 Schéma trénovací smyčky neuronové sítě. 11
2.1 Schéma dopředného a zpětného difuzního procesu. 17
3.1 Spektrogram přelaďovaného harmonického signálu s chybějící částí. . 21
3.2 Průběh sinusového signálu zkresleného clippingem. 22
4.1 Spektrogram signálu uměle zkresleného clippingem s thresholdem 0,2. 25
4.2 Časový průběh signálu uměle zkresleného clippingem s thresholdem 0,2. 25
4.3 Spektrogram signálu uměle zkresleného clippingem s thresholdem 0,25. 28
4.4 Evaluace rekonstruovaného signálu pomocí Diff SDR. 29
4.5 Evaluace rekonstruovaného signálu pomocí ℓ_1 normy. 30
4.6 Evaluace rekonstruovaného signálu pomocí ℓ_2 normy. 31
4.7 Porovnání výstupů použitých metrik. 31

Úvod

Tato práce se zabývá tématem difuzních modelů neuronových sítí a jejich využití pro opravu audio signálů poškozených clippingem.

Neuronové sítě využívají takzvané hluboké strojové učení. Jsou předmětem výzkumu a vývoje již dlouhou řadu let, přičemž v současné době se této problematice dostává více pozornosti. Vzrůstající oblibě se těší zejména difuzní modely, jejichž aplikace přináší čím dál kvalitnější výsledky při generování dat, proto mají velký potenciál i pro úlohu rekonstrukce poškozeného audio signálu.

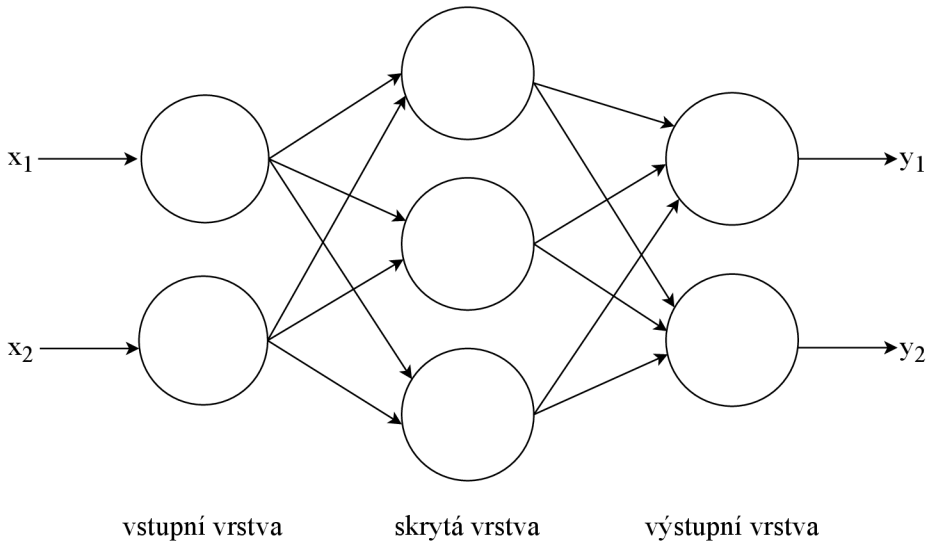
Cílem této bakalářské práce je nastudovat teorii problematiky, implementovat řešení navržené v článku [1] a navrhnout modifikaci, která by jej mohla vylepšit, případně umožnit využití v jiné rekonstrukční úloze. Uvedený zdroj popisuje způsob využití difuzních modelů neuronových sítí pro opravu audio signálů postihnutých tzv. inverzními problémy, mezi něž patří audio inpainting, declipping, či zvětšení šířky pásma. V rámci této práce se budeme soustředit pouze na audio declipping, tedy na rekonstrukci signálu poškozeného clippingem. Vybrané řešení je realizováno v jazyce Python s využitím frameworku PyTorch [2], který je v této oblasti aktuálně k jedním z nejvyužívanějších a je k němu dostupná kvalitní dokumentace.

První část této práce seznamuje čtenáře s tematikou neuronových sítí, strojového učení a se základními pojmy, nezbytnými pro pochopení jejich architektury a funkce. V druhé části je vysvětlen princip difuzních modelů neuronových sítí s výkladem nejdůležitějších matematických principů, které jsou jejich základem. Ve třetí části je čtenář uveden do problematiky inverzních problémů v oblasti audio signálů a seznámen s jejich konkrétními příklady. Poslední část textu popisuje, jakých výsledků se při programové realizaci zvolené metody podařilo dosáhnout, výstupy algoritmu jsou závěrem zhodnoceny pomocí objektivních metrik.

1 Neuronové sítě

Neuronové sítě jsou modelem pro uplatnění strojového učení, které využívá umělá inteligence. Koncept neuronových sítí je inspirován strukturou hustých a nesčetných propojení neuronů v lidském mozku, z čehož vychází i samotné pojmenování. [3]

V rámci strojového učení rozlišujeme hluboké a mělké učení, přičemž hlavní rozdíl mezi nimi je v počtu vrstev, z nichž síť sestává [4]. V této práci se budeme zabývat difuzními modely neuronových sítí, které využívají hluboké učení.

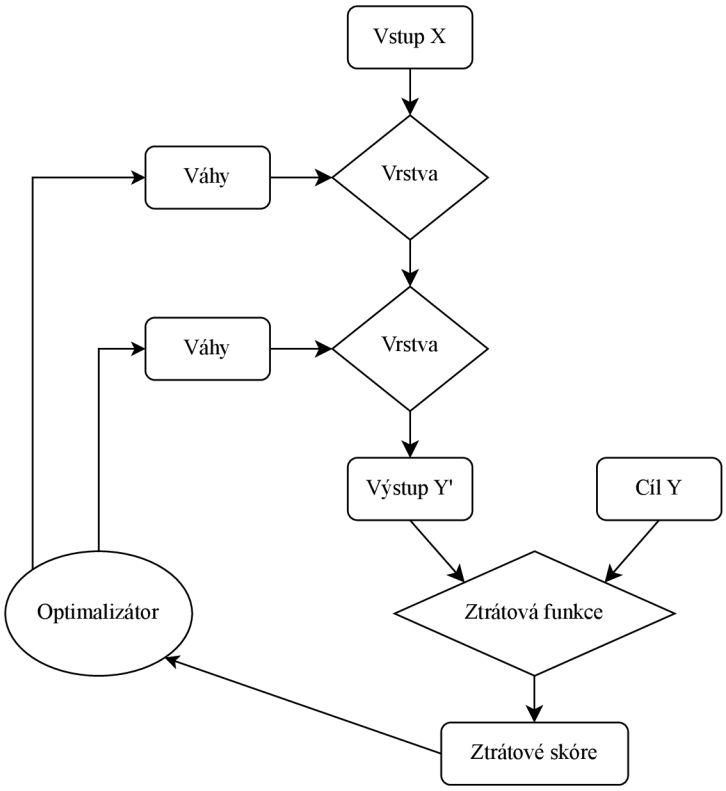


Obr. 1.1: Schéma propojení vrstev neuronové sítě.

Architektura neuronové sítě sestává z vzájemně propojených vrstev, jejichž úkolem je zpracovávat přijatá data a výsledek předat dál. Tato struktura je zjednodušeně ilustrována na obrázku 1. Každá vrstva je definována parametry zvanými váhy (weights). Těmito parametry jsou jádro (kernel) a zkreslení (bias) [4]. Jedná se o číselné hodnoty přiřazené vstupním datům, které udávají, jakým způsobem budou tato data danou vrstvou transformována. [3]

Hodnoty parametrů vrstvy jsou následně upravovány v rámci trénování sítě, jehož cílem je postupně transformovat vstupní data tak, aby se jejich výstupní hodnoty co nejvíce přiblížily těm očekávaným. Míru, do jaké se to podaří, sleduje ztrátová funkce. Ta porovná hodnoty predikce a skutečného výsledku a ztrátu vyčíslí pomocí ztrátového skóre. Toto skóre je žádoucí co nejvíce snížit za pomoci optimalizátoru, který na základě výpočtu gradientů v rámci trénovaného modelu sítě provádí aktualizaci vah podle ztrátové funkce. Celý proces je součástí trénovacího

cyklu znázorněného na obrázku 1.2, který během mnohonásobného opakování nastavuje hodnoty vah tak, aby byla ztráta co nejnižší. Dosažením co nejnižší možné ztráty, kdy se výstupy co nejvíce blíží predikcím, získáme natrénovanou síť. [4]



Obr. 1.2: Schéma trénovací smyčky neuronové sítě.

1.1 Tenzory

Neuronové sítě pracují s daty ve formátu tenzoru. Jedná se o způsob uložení dat do vícedimenzionálního pole. Počet jeho dimenzí je dán počtem os tohoto pole. Skalár lze považovat za 0D tenzor — obsahuje pouze jedno číslo, nemá tedy žádnou osu. Jednodimenzionální tenzor je potom vektor, neboť má právě jednu osu, 2D tenzor je matice, ta má osy dvě. Tenzory, s nimiž neuronové sítě pracují, mohou mít například tři, čtyři, případně více dimenzí. Obvykle se však nepracuje s tenzory, které by měly více než pět dimenzí — to je případ modelů zpracovávajích videa. [4]

1.2 Vrstvy

Vrstva je základním stavebním blokem neuronové sítě. Jejím úkolem je zpracovat jeden či více tenzorů, které přijímá na vstupu, a jako výsledek opět vrátit na výstupu jeden či více tenzorů. V průběhu tohoto procesu síť získává ze vstupních dat tzv. reprezentace, tedy v podstatě jiný typ informace, který sice zastupuje původní data, ale v nové podobě je daleko snáze zpracovatelný pro naši síť, což výrazně usnadňuje její trénování. [4]

Reprezentací pro audio signál může být například magnitudový spektrogram, který v rámci dvou os – horizontální pro čas, vertikální pro frekvenci – vyobrazuje, jakého výkonu zvuk dosahuje na kterých frekvencích v průběhu času. Je však třeba podotknout, že komplexní spektrogram zobrazuje nejen výkon, ale také fázovou reprezentaci signálu. Pokud tedy zvukový signál v podobě tenzoru přetransformujeme do podoby spektrogramu, získáme reprezentaci, která je již na první pohled přehlednější a srozumitelnější a zároveň nám dá informaci o harmonickém složení signálu v čase. [5]

Neuronová síť obvykle sestává ze vstupní vrstvy, jedné nebo vícero skrytých vrstev, a z jedné výstupní vrstvy [4]. Existuje mnoho typů vrstev, jejich využití se liší jednak podle formátu tenzoru, s nímž má neuronová síť pracovat, a jednak podle způsobu, jakým mají být data zpracována. Níže je popsáno několik vybraných typů vrstev: plně propojené, konvoluční, rekurentní, vrstvy s dlouhou krátkodobou pamětí (LSTM) a výpadkové (dropout) vrstvy. [6]

Plně propojené vrstvy, rovněž nazývané hustě propojené či lineární, (dense, fully connected) aplikují na vstupní tenzor lineární transformaci, která ve své podstatě spočívá v aplikaci samotných vah vrstvy dle rovnice

$$\mathrm{y} = \mathrm{x}\mathrm{W}^{\mathrm{T}} + \mathrm{b}, \tag{1.1}$$

přičemž y označuje výstup a x je vstupní signál. Ten je podroben nejprve maticovému součinu s maticí vah W, označující jádro (kernel) a následně součtu s b, označujícím zkreslení (bias). Název tohoto typu vrstvy vychází ze skutečnosti, že neuron v plně propojené vrstvě přijímá výstup ze všech neuronů vrstvy předcházející. [7, 8]

Konvoluční vrstvy jsou základem konvolučních neuronových sítí a jsou hojně využívány při zpracování obrázků i zvuku. Tento typ vrstvy provádí konvoluci tak, že posouvá okno o daných rozměrech po mapě příznaků (3D tenzor) a provede součin po jejích složkách v tomto okně s tzv. konvolučním jádrem. Výstupem je rovněž mapa příznaků, její rozměry se od vstupního tenzoru liší především v hloubce. [4]

Rekurentní vrstvy zpracovávají sekvenční data. Jejich vnitřní vrstva se v průběhu času vyvíjí, což jim umožňuje zachycovat informace z předchozích kroků. Tento typ vrstev se využívá např. při zpracování řeči. [6]

Vrstvy typu LSTM (Long Short-Term Memory), v překladu vrstvy s dlouhou krátkodobou pamětí, jsou zvláštním typem rekurentní vrstvy, které dokážou obejít problém mizejícího gradientu algoritmu zpětného šíření chyby u rekurentních neuronových sítí. Tato vrstva využívá vstupní bránu, výstupní bránu, bránu zapomínání a vnitřní buňku k regulaci toku informací neuronovou sítí, což jí poskytuje onu dlouhou krátkodobou paměť. [6]

Výpadkové, neboli dropout vrstvy vynulují určitý počet náhodně vybraných neuronů, čímž se snaží zabránit tomu, aby se síť příliš spoléhala na jednotlivé neurony. [6]

1.3 Aktivační funkce

Mezi některými vrstvami neuronové sítě je třeba aplikovat i tzv. aktivační funkci, která u jednotlivých neuronů rozhoduje o tom, zda budou aktivovány. Jedná se o matematickou funkci aplikovanou na jejich výstupu, která v modelu umožňuje zachycení a zpracování nelineárních jevů, právě tak jako uplatňování i jiných než lineárních operací s tenzory. Tato funkce je důležitá pro to, aby se síť dokázala učit komplexní vzorce ve zpracovávaných datech a získávat z nich potřebné reprezentace. Také umožňuje síti provádět algoritmus zpětného šíření tím, že poskytne vypočtené gradienty, nezbytné pro aktualizaci vah. [4, 9]

K běžně používaným typům aktivační funkce patří sigmoid, hyperbolický tangens a ReLU, existují také exponenciální aktivační funkce typu softmax a softplus. [9]

1.4 Gradient

Gradient je derivací funkce, která má více než jednu vstupní proměnnou [10]. Je definován jako vektorové pole, které vyjadřuje směr a velikost změny skalárního pole [11]. Gradient skalární funkce dvou proměnných lze popsat následující rovnicí:

$$\nabla f = \left(\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y} \right). \tag{1.2}$$

V oblasti neuronových sítí označuje gradient derivaci tenzorové operace a je zásadním elementem v optimalizačním procesu. Neuronová síť však musí nejprve vypočítat gradient ztráty předtím, než dojde k samotné optimalizaci [4]. Toto platí v rámci funkce spojité, nikoliv diskrétní, v níž se budeme v rámci praktické části této práce pohybovat. PyTorch má však nástroje, umožňující výpočet i v této doméně.

1.5 Optimalizace

Optimalizace úzce souvisí se ztrátovou funkcí. Jejím úkolem je aktualizovat míru zkreslení a váhy v průběhu trénování sítě tak, aby byla ztráta s každým krokem minimalizována co nejvíce. [4]

V rámci trénování probíhá optimalizace ve třech krocích: nejprve jsou parametry modelu neuronové sítě inicializovány na náhodné hodnoty, následně je zpětným průchodem sítí vyčíslena očekávaná ztráta, a nakonec dojde k úpravě parametrů na základě porovnání výstupů [12]. Tomuto procesu rovněž můžeme říkat gradientní sestup. Spočívá nejprve ve výpočtu gradientu ztrátové funkce ve vztahu k váhám, které jsou následně upraveny v opačném směru gradientu. [13]

I optimalizace má své parametry, které je třeba vhodně nastavit před trénováním sítě. K nim patří počet epoch, velikost dávky (batch) a rychlost učení. Jedna epocha znamená jednu iteraci na trénovacím datasetu, počet epoch tedy udává, kolik iterací síť provede. Velikost dávky definuje počet datových vzorků, které projdou sítí v rámci jedné epochy. Rychlost učení nastavuje, o jak velkou hodnotu mohou být váhy aktualizovány po průběhu jedné epochy. [12]

Často používanými typy optimalizátorů jsou gradientní sestup, stochastický gradientní sestup, AdaGrad, Adadelta, RMSProp, či Adam optimizer. Optimalizátor založený na gradientním sestupu upravuje váhy ve směru nejprudšího poklesu ztrátové funkce. Základní myšlenkou tohoto principu je následovat pokles ztráty, abychom se co nejvíce přiblížili jejímu minimu. V případě stochastického gradientního sestupu (SGD) je princip podobný, gradient zde však není vypočítán pro celý trénovací dataset, nýbrž pro každý trénovaný vzorek zvlášť. Optimalizátor AdaGrad přizpůsobuje rychlost učení tak, aby pro častěji aktualizované parametry byla nižší, a pro méně často aktualizované naopak vyšší. RMSProp optimalizátor svou rychlost učení přizpůsobuje průměru posledních gradientů pro daný parametr. Adam optimizer je kombinací optimalizátorů typu AdaGrad a RMSProp, svou rychlost učení upravuje dynamicky v průběhu trénování. Tento optimalizátor má uplatnění v široké škále optimalizačních problémů, v hlubokém učení je hojně využívaný. [14]

1.6 Ztrátová funkce

Ztrátová funkce nám umožňuje sledovat, do jaké míry se skutečný výstup neuronové sítě blíží požadovanému výsledku. Tato funkce za pomoci předem zvoleného mechanismu porovná odchylku mezi očekávaným a reálným výstupem a následně vyčíslí ztrátu. Cílem je zajistit, aby toto číslo bylo co nejnižší. Neuronová síť může mít více než jednu ztrátovou funkci. Má-li síť více výstupů, může mít pro každý z nich samostatnou ztrátovou funkci. Ztrátové funkce je také možno sčítat a váhovat. [4]

Jakým mechanismem bude porovnán skutečný výstup s očekávaným, udává typ ztrátové funkce, který je proto třeba vhodně zvolit dle řešeného problému. Pro regresní problémy se například používá střední kvadratická chyba (MSE) nebo střední absolutní chyba (MAE). Pro klasifikaci je vhodná binární křížová entropie nebo kategorická křížová entropie. Typů ztrátových funkcí je nepřeberně mnoho a vzájemně se liší ve svém využití, výše uvedené však patří k nejčastěji používaným. [4]

2 Difuzní neuronové sítě

Difuzní modely pracují na principu zkreslování vstupního signálu postupným přidáváním Gaussova šumu až do fáze, kdy je signál zcela zašuměn a ztrácí jakoukoliv podobnost se vstupním vzorkem. Následuje proces postupného odstraňování šumu dokud není signál obnoven v původní nezkreslené podobě. První zmíněný proces se nazývá dopředný. Druhý proces se nazývá reverzní, případně zpětný, a jeho účelem je naučit neuronovou síť, aby ze zcela zašuměného vzorku rekonstruovala signál do podoby původního nezkresleného signálu. [15]

Tento typ neuronových sítí spadá do kategorie pravděpodobnostních generativních modelů [16]. Účelem generativních modelů je vytváření dat podobných těm, na kterých se daná neuronová síť učí. V případě difuzních modelů je cílem síť natrénovat tak, aby následně dokázala vytvářet data z realizace Gaussovského šumu. Tato metoda je již využívána pro generování např. obrázků – v současné době sklízí úspěchy model DALL-E 2 [17], který podobně jako většina dalších populárních komerčních modelů od Open AI (SORA, Stable Diffusion a další) úspěšně využívá difuzní neuronové sítě. [18]

Matematicky lze celý difuzní proces vyjádřit pomocí Markovova řetězce. Nejprve je ale třeba upřesnit, co je to Markovův řetězec. Zdroj [19] jej popisuje jako „náhodný proces bez paměti", jehož budoucí stav závisí pouze na stavu současném, ale nikoliv na minulém. Tento proces pozoruje jakousi stavovou soustavu v okamžicích $n = 0, 1, ...$ Změnu o jeden časový okamžik, tedy změnu z okamžiku n na okamžik $n + 1$, lze uvažovat jako přechod za jeden krok. Takto popsaný systém nemá paměť, neboť i kdybychom znali jeho minulý stav, stav budoucí bude stále záviset pouze na tom přítomném. [19]

Abychom s využitím této metody mohli popsat princip difuzních modelů, jsou definovány dvě podoby Markovova řetězce. Nejprve je uměle navržen dopředný Markovův řetězec, který přetvoří data na šum v rámci T kroků [16]. Jednotlivý krok je definován vzorcem

$$q(\mathrm{x}_n|\mathrm{x}_{n-1}), \tag{2.1}$$

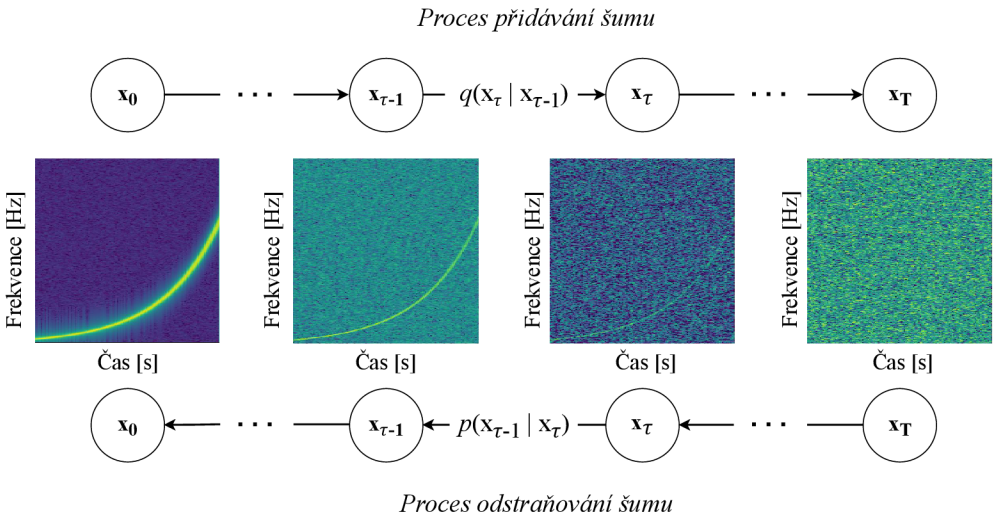
kde x_n označuje současný stav a x_{n-1} stav budoucí [20]. Během T kroků je postupně k aktuálnímu vzorku přičítán Gaussovský šum, počínaje nezkresleným vzorkem vstupního signálu x_0, konče zcela zašuměným vzorkem x_T. Takto získáme celou sekvenci zašuměných vzorků x_1,...,x_T. S každým dalším krokem je zašuměný signál čím dál méně podobný původnímu vzorku x_0. Nakonec pro T blížící se nekonečnu je vzorek x_T ekvivalentem izotropního Gaussova rozdělení. [15]

Výsledný šum je následně zpracován zpětným Markovovým řetězcem prostřednictvím parametrizace jader tranzice (transition kernels) neuronovou sítí [16].

Zpětný Markovův řetězec je definovaný jako

$$p(\mathrm{x}_{n-1}|\mathrm{x}_n), \tag{2.2}$$

přičemž x_{n-1} značí aktuální stav a x_n je stav následující. Síť je trénována tak, aby šum odstranila a obnovila původní signál bez zkreslení. Ve své podstatě však vytváří zcela nový signál z pouhého šumu. [20]



Obr. 2.1: Schéma dopředného a zpětného difuzního procesu.

Markovův řetězec popisuje difuzní proces v diskrétním čase. Lze jej však popsat i v čase spojitém, a to jako stochastický proces. Dopředný proces je pak v tomto pojetí souborem náhodných proměnných $\{\mathrm{x}_\tau\}_{\tau\in[0,\mathrm{T}]}$ s indexem $\tau \in [0,\mathrm{T}]$, označujícím spojitý čas zpracování. Toto τ u stochastických procesů koresponduje s indexem n u Markovových řetězců. [20]

Jádro tranzice $q_\tau(\mathrm{x}_\tau|\mathrm{x}_0)$ představuje podmíněné pravděpodobnostní rozdělení, které charakterizuje dopředný difuzní proces a souvisí s pravděpodobností v Markovových řetezcích $q(\mathrm{x}_n|\mathrm{x}_0)$. Lze jej popsat stochastickou diferenční rovnicí, tj. diferenční rovnici, v níž jsou některé koeficienty náhodné. Podoba dopředné stochastické diferenční rovnice je následující:

$$\mathrm{dx}_\tau = \mathrm{f}(\mathrm{x}_\tau, \tau)\mathrm{d}\tau + g(\tau)\mathrm{dw}_\tau. \tag{2.3}$$

Zde vektorová funkce f značí driftový koeficient, který představuje deterministickou část rovnice. Skalární funkce g značí difuzní koeficient, který reguluje míru

náhodnosti ve stochastické diferenční rovnici. Koeficient $g(\tau)$ reguluje šum vnesený stochastickým procesem w_τ, za který se obvykle dosazuje Wienerův proces, tedy stochastický proces s nezávislými a normálně rozdělenými přírůstky. Jádro tranzice může také nabýt jednoduché Gaussovské podoby vyjádřené rovnicí uvedenou níže za předpokladu, že driftový koeficient f bude afinní funkcí x_τ a difuzní koeficient g bude nezávislý na x_τ:

$$q_\tau(\mathrm{x}_\tau|\mathrm{x}_0) = \mathcal{N}(\mu(\mathrm{x}_0, \tau), \sigma(\tau)^2\mathbf{I}). \tag{2.4}$$

Zde hodnoty odhadu střední hodnoty $\mu(\mathrm{x}_0, \tau)$ a standardní odchylky $\sigma(\tau)$ vypočítáme numerickým řešením stochastické diferenciální rovnice, přičemž standardní odchylka v podobě $\sigma(\tau)^2$ zde v kombinaci s jednotkovou maticí $\mathbf{I}$ udávají rozptyl. [20]

Zpětný difuzní, respektive generativní proces je rovněž stochastickým procesem x_τ, přičemž doba zpracování $\tau \in [0,\mathrm{T}]$ plyne opačným směrem. Obrátíme-li časovou osu, získáme zpětnou (reverzní) stochastickou diferenciální rovnici, přičemž proces odšumění vzorku x_τ spočívá v jejím vyřešení:

$$\mathrm{dx}_\tau = \left[\mathrm{f}(\mathrm{x}_\tau, \tau) - g(\tau)^2 \nabla_{x_\tau} \log p_\tau(\mathrm{x}_\tau)\right] \mathrm{d}\tau + g(\tau)\mathrm{d}\bar{\mathrm{w}}_\tau. \tag{2.5}$$

V tomto vztahu $\bar{\mathrm{w}}$ označuje Wienerův proces, realizovaný v rámci obrácené časové osy – z hlediska provedení se tedy nejedná o tentýž Wienerův proces jako u dopředného procesu. Je-li definováno dw_τ pro dopředný proces, pak jej nelze zaměnit s $\mathrm{d}\bar{\mathrm{w}}_\tau$ pro proces zpětný. Gradient logaritmické funkce se zde nazývá funkce skóre (score function) a nabývá podoby vektorového pole, které nese informace o hustotě logaritmické pravděpodobnosti modelu. Aproximace tohoto skóre je možná za pomoci modelu skóre (score model), který označujeme $s_\theta(\mathrm{x}_\tau, \tau)$. Tento model skóre je výhodný, neboť na rozdíl od klasického rozdělení nepotřebuje být normalizovaný, plocha pod grafem se rovná jedné. Tento model můžeme optimalizovat porovnáváním skóre odšumění (denoising score matching), přičemž namísto skóre neznámé pravděpodobnosti $p_\tau(\mathrm{x}_\tau)$ zde můžeme k porovnávání využít Gaussovské jádro tranzice $q_\tau(\mathrm{x}_\tau|\mathrm{x}_0)$, které lze ze vztahu 2.4 vyjádřit následující rovnicí:

$$\nabla_{\mathrm{x}_\tau} \log q_\tau(\mathrm{x}_\tau|\mathrm{x}_0) = -\frac{\mathrm{x}_\tau - \mu(\mathrm{x}_0, \tau)}{\sigma(\tau)^2}. \tag{2.6}$$

Toto porovnávání skóre je součástí trénování modelu skóre, v jehož rámci je nejprve zvolen vzorek x_0 jakožto datový bod v trénovacím datasetu a difuzní čas τ v rozmezí od 0 do T. Stav x_τ lze následně získat vzorkováním z jádra tranzice $q_\tau(\mathrm{x}_\tau|\mathrm{x}_0)$ dle níže uvedeného vztahu:

$$\mathrm{x}_\tau = \mathrm{x}_0 + \sigma(\tau)\mathcal{N}(0, 1). \tag{2.7}$$

Po natrénování je model skóre schopen vzorkovat signál na základě naučeného rozdělení dat numerickým řešením zpětné stochastické diferenciální rovnice 2.5. [20]

2.1 Podmíněné a nepodmíněné modely

Difuzní modely lze aplikovat dvěma způsoby, a to jako podmíněné a nepodmíněné modely. Podmíněné byly ze zmíněných dvou vyvinuty později, teprve na základě modelů nepodmíněných. Na rozdíl od těch nepodmíněných umožňují regulovat výsledky generativního procesu v souladu s naším záměrem. To probíhá za pomoci podmínek, zadaných prostřednictvím textu, obrázků, grafů, klasifikátorů či sémantických map, atd. Průběh samotného generativního procesu má zde stejnou podobu jako u modelů nepodmíněných, za pomoci oněch podmínek je však směrován tak, aby se výsledná vygenerovaná data co nejvíce podobala stanovenému cíli. [16]

Způsoby podmíněného generování mohou být různé, některé vyžadují aplikaci dodatečného vstupního signálu, který vychází ze stejné datové instance jako hlavní vstupní signál – zde může být hlavním signálem čistý audio signál a dodatečným jeho poškozená verze. Jiný způsob může být přizpůsobení již samotných parametrů difuzního procesu danému problému, který má být vyřešen. Dalším způsobem může být externí podmínění, kdy model samotný je natrénován jako nepodmíněný, přičemž teprve při jeho testování (inferenci) je aplikován podmiňující signál. Existují také postupy, které kombinují výše uvedené metody. [20]

V rámci této práce se budeme zabývat řešením inverzních úloh pomocí nepodmíněného modelu, který je možné využít pro různé úlohy bez nutnosti model opětovně trénovat pro každou z nich zvlášť.

3 Inverzní úlohy

Inverzními úlohami, nebo též inverzními problémy, v oblasti audio signálů rozumíme takové úlohy, při jejichž řešení usilujeme o zvrácení příčiny poškození signálu. V minulosti se to týkalo audio signálů, u nichž nám byl způsob poškození předem znám, což už v současné době ne vždy platí. V rámci této práce však budeme řešit situaci, kdy nám operátor poškození znám bude. Do kategorie těchto úloh spadá například zvětšení šířky pásma (bandwidth extension), doplnění chybějícího úseku signálu (inpainting), případně také declipping. [1]

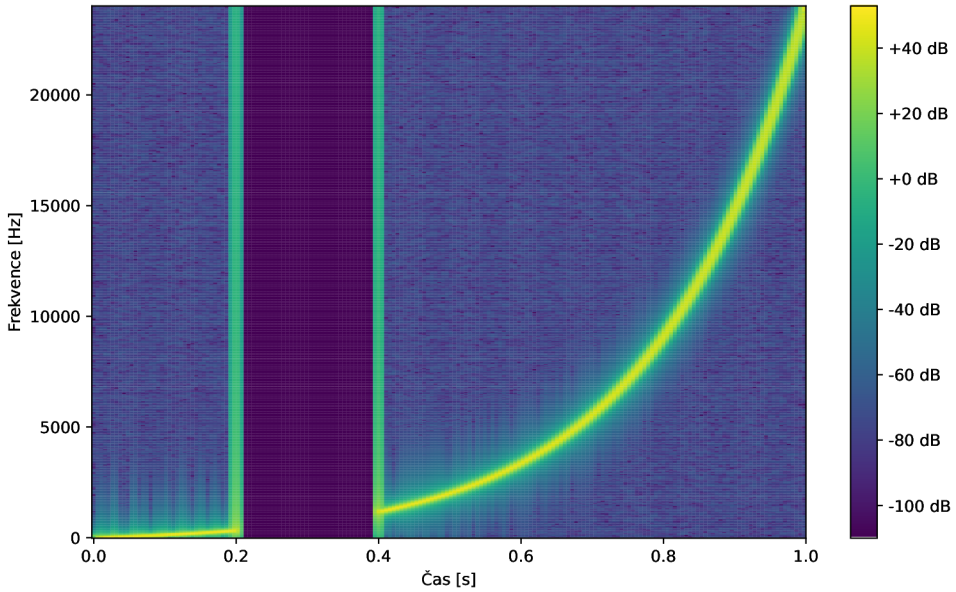
V této části práce si nejprve představíme princip jednotlivých inverzních problémů a následně popíšeme jejich možné řešení v kontextu difuzních modelů, a sice realizací difuzního aposteriorního vzorkování.

3.1 Zvětšní šířky pásma

Šířka pásma (bandwidth) je rozdíl mezi nejvyšší a nejnižší frekvencí přenášeného signálu [21]. Její rozšíření bývá žádoucí nejčastěji v případě komprese audio signálu. U takovýchto signálů se obvykle usiluje o doplnění chybějících vyšších frekvencí, které byly potlačeny, špatně zaznamenány, či jiným způsobem ztraceny, případně nebyly zaznamenány vůbec. Především v případě mluveného slova pak jejich nedostatek snižuje srozumitelnost, u zvuku obecně jasnost, dokonce může vést i ke ztrátě informací. Tento problém má velkou váhu zejména v oblasti komunikačních technologií. Rozšiřování šířky pásma dále nalezlo uplatnění i v případě automatického rozpoznávání řeči, syntézy řeči z textu, hlasových, resp. řečových kodeků, zvyšování kvality historických zvukových nahrávek, a v dalších oblastech. [22]

3.2 Doplnění chybějící části signálu

Audio inpainting je metoda, jejímž cílem je doplnění chybějící části signálu, případně obnova části signálu natolik poškozené, že její oprava již není možná. Situace, kdy v signálu chybí vzorky, či jsou poškozeny, může nastat například při digitalizaci audio záznamu z fyzických nosičů, které utrpěly mechanické poškození – typickým příkladem je poškrábané CD. Dalším příkladem takové situace může být ztráta paketů (packet loss) při přenosu dat v komunikační síti. Prostřednictvím audio inpaintingu lze ze signálu také odstraňovat šum, či jiné nežádoucí jevy (artefakty). V neposlední řadě může být tato metoda kreativním nástrojem v hudební produkci například pro tvorbu zvukových efektů. [23]



Obr. 3.1: Spektrogram přelaďovaného harmonického signálu s chybějící částí.

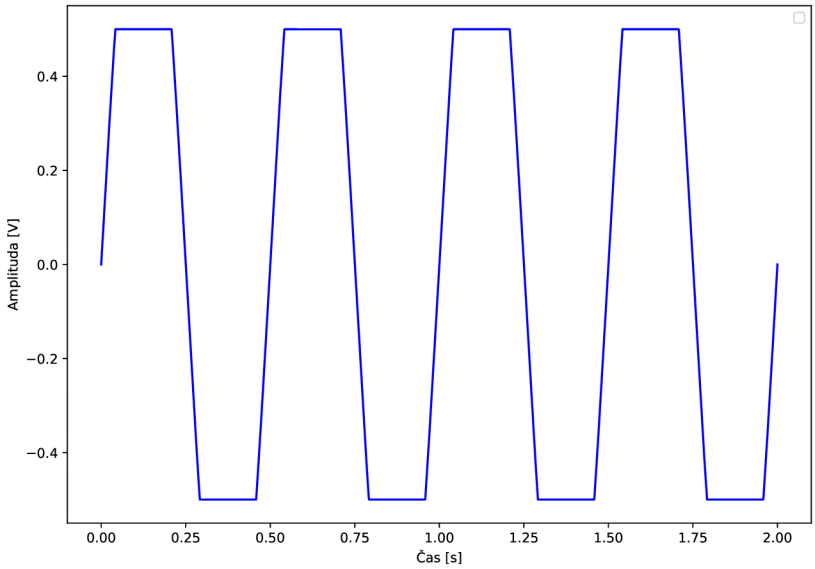
Pro vizualizaci signálu s chybějící částí je přiložen spektrogram uměle poškozeného přelaďovaného harmonického signálu 3.1, jehož frekvence se plynule zvyšuje v průběhu času od počátečních 20 Hz do konečných 20 kHz. Spektrogram je způsob zobrazení časového průběhu audio signálu, přičemž v závislosti na barvě podává informace o úrovni signálu na jednotlivých frekvencích v decibelech. Barevné značení může být různé — to, které je použité zde, uvádí tmavě modrou barvu jako označení pro slabý neslyšitelný signál dosahující −100 dB a méně. Škála dále přechází od nejtišších jevů k hlasitějším přes mírně světlejší modrou a zelenou barvu, přičemž nejhlasitější jevy, dosahující +40 dB a více jsou značeny žlutou barvou.

3.3 Declipping

Takzvaný clipping, někdy též nazýván saturace, je typ nelineárního poškození audio signálu. Tento jev nastává v případě přebuzení zvukového signálu nad rámec povoleného dynamického rozsahu například vlivem příliš vysokého nastavení vstupního zisku (gainu) signálu při jeho nahrávání. Někdy sice může být vyvoláván záměrně pomocí zvukových efektů typu overdrive či distortion pro dosažení specifického estetického záměru. V ostatních případech však bývá nežádoucí, a nepodaří-li se mu předejít již před záznamem signálu, je obtížné jej odstranit zpětně. Metoda odstranění clippingu se nazývá declipping. [24, 25]

Rozlišujeme dva druhy clippingu: hard clipping a soft clipping. Soft clipping se projevuje zaoblením signálu v místě, kde se jeho amplituda blíží prahu (thresholdu)

povoleného dynamického rozsahu. Hard clipping se projevuje ostrým ořezáním amplitudy signálu v místě, kde práh překračuje. [26] Tento případ je vyobrazen v níže přiloženém grafu časového průběhu zkresleného harmonického signálu 3.3.



Obr. 3.2: Průběh sinusového signálu zkresleného clippingem.

3.4 Difuzní aposteriorní vzorkování

Difuzní modely mají za cíl odhadovat apriorní podobu výchozího rozdělení dat za pomoci gradientu logaritmické hustoty $\nabla_{\mathrm{x}} \log p(\mathrm{x})$. Tato funkce nachází využití v řešení inverzních úloh, kdy usilujeme o obnovení signálu x na základě změřených hodnot y vzorkováním z Gaussovského šumu. Vztah mezi x a y závisí na operátoru $\mathcal{A}$, kterým je dopředný model, a na Gaussovském šumu n dle vzorce $\mathrm{y} = \mathcal{A}(\mathrm{x}_0) + n$. [27]

Aposteriorní rozdělení lze popsat pomocí Bayesovy věty:

$$p(\mathrm{x}|\mathrm{y}) = \frac{p(\mathrm{y}|\mathrm{x})p(\mathrm{x})}{p(\mathrm{y})}, \tag{3.1}$$

kde $p(\mathrm{x})$ označuje apriorní pravděpodobnostní rozdělení x bez podpůrných dat y. Dále $p(\mathrm{y}|\mathrm{x})$ označuje pravděpodobnost jevu y, podmíněnou parametrem x. Pravděpodobnost jevu y značená $p(\mathrm{y})$ představuje mezní pravděpodobnost a používá se jako normující konstanta. [28]

V kontextu inverzních úloh je cílem zjistit x_0, tedy původní podobu signálu, na základě jeho podoby x, tedy poté, co byl signál například poškozen anebo nekvalitně zaznamenán. Celé toto aposteriorní pravděpodobnostní rozdělení dat $p(\mathrm{x}_0|\mathrm{y})$

nelze explicitně vypočítat, namísto toho je však možné odhadnout jeho střed pomocí Tweedieho vzorce:

$$\hat{\mathrm{x}}_0 := \mathbb{E}[\mathrm{x}_0|\mathrm{x}_\tau] = \frac{1}{\sqrt{\bar{\alpha}(\tau)}}(\mathrm{x}_\tau + (1-\bar{\alpha}(\tau))\nabla_{\mathrm{x}_\tau} \log p_\tau(\mathrm{x}_\tau)), \tag{3.2}$$

známe-li gradient hustoty dat $\nabla_{\mathrm{x}_\tau} \log p_\tau(\mathrm{x}_\tau)$, který lze nahradit funkcí skóre $s_\theta(\mathrm{x}_\tau, \tau)$. Zde $(1-\bar{\alpha}(\tau))$ představuje rozptyl σ^2. Tento odhad nepoškozené podoby signálu lze následně využít k odhadu způsobu, jakým k poškození došlo. Základem aposteriorního vzorkování je níže uvedený vztah:

$$\nabla_{\mathrm{x}_\tau} \log p_\tau(\mathrm{x}_\tau|\mathrm{y}) = \nabla_{\mathrm{x}_\tau} \log p_\tau(\mathrm{y}|\mathrm{x}_\tau) + \nabla_{\mathrm{x}_\tau} \log p_\tau(\mathrm{x}_\tau), \tag{3.3}$$

kterým nahrazujeme nepodmíněnou část rovnice 2.5, čímž zajistíme vedení generativního procesu nejpravděpodobnějším směrem. Gradient $\nabla_{\mathrm{x}_\tau} \log p_\tau(\mathrm{y}|\mathrm{x}_\tau)$ zde můžeme aproximovat následovně:

$$\nabla_{\mathrm{x}_\tau} \log p_\tau(\mathrm{y}|\mathrm{x}_\tau) \simeq \nabla_{\mathrm{x}_\tau} \log p_\tau(\mathrm{y}|\hat{\mathrm{x}}_0). \tag{3.4}$$

Odhad gradientu hustoty lze poté vypočítat pomocí ℓ_2 normy jako

$$\nabla_{\mathrm{x}_\tau} \log p_\tau(\mathrm{y}|\hat{\mathrm{x}}_0) = -\rho \nabla_{\mathrm{x}_\tau} \|\mathrm{y} - \mathcal{A}(\hat{\mathrm{x}}_0)\|_2^2, \tag{3.5}$$

kde ρ je velikost kroku, definovaná jako $\frac{1}{\sigma^2}$. Dosazením výše uvedeného do vzorce 3.3 získáme rovnici Gaussovského difuzního aposteriorního vzorkování, která je klíčem k řešení inverzních úloh:

$$\nabla_{\mathrm{x}_\tau} \log p_\tau(\mathrm{x}_\tau|\mathrm{y}) = s_{\theta^*}(\mathrm{x}_\tau, \tau) - \rho \nabla_{\mathrm{x}_\tau} \|y - \mathcal{A}(\hat{\mathrm{x}}_0)\|_2^2. \tag{3.6}$$

Za předpokladu, že známe dopředný model $\mathcal{A}$, je následně možné realizovat ono vzorkování z aposteriorního rozdělení. [27]

4 Výsledky

V této části práce bude prezentováno programové řešení spolu s výsledky, jejichž kvalita bude následně posouzena na základě objektivních metrik. Kód práce je přístupný ve veřejném GitHub repozitáři: `https://github.com/247800/DPSampler`.

4.1 Programové řešení

Zdroj [1] doporučuje řešení inverzních úloh za pomoci difuzních modelů s využitím tzv. CQT (Constant-Q Transformation), tedy transformace s konstantním Q. Tato metoda pracuje s logaritmickou frekvenční osou, což ji činí výhodnou pro práci se zvukovým signálem, a to především s hudbou. Dle uvedeného zdroje se prokázala být velmi efektivním nástrojem při řešení různých typů inverzních úloh (zvětšení šířky pásma, inpainting, declipping) bez nutnosti trénovat tentýž model pro každou z těchto úloh zvlášť.

Řešení bylo implementováno v jazyku Python 3.10.11 s využitím frameworku PyTorch 2.6.0 [2], který má veřejně dostupnou detailně popsanou dokumentaci, zároveň patří v současné době k nejvyužívanějším open source nástrojům pro vývoj umělé inteligence, pročež je k němu dostupné hojné množství informativních materiálů.

4.1.1 Dataset

Při návrhu algoritmu pro realizaci aposteriorního vzorkování byl použit dataset IDMT-SMT-GUITAR [29], obsahující několikasekundové monofonní nahrávky tří různých kytar, hrající v různých tempech a žánrech. Nahrávky jsou ve formátu RIFF WAVE, s vzorkovací frekvencí 44,1 kHz. Pro účely této práce byla konkrétně zvolena sada 64 jazzových nahrávek kytary s bitovou hloubkou 16 bitů.

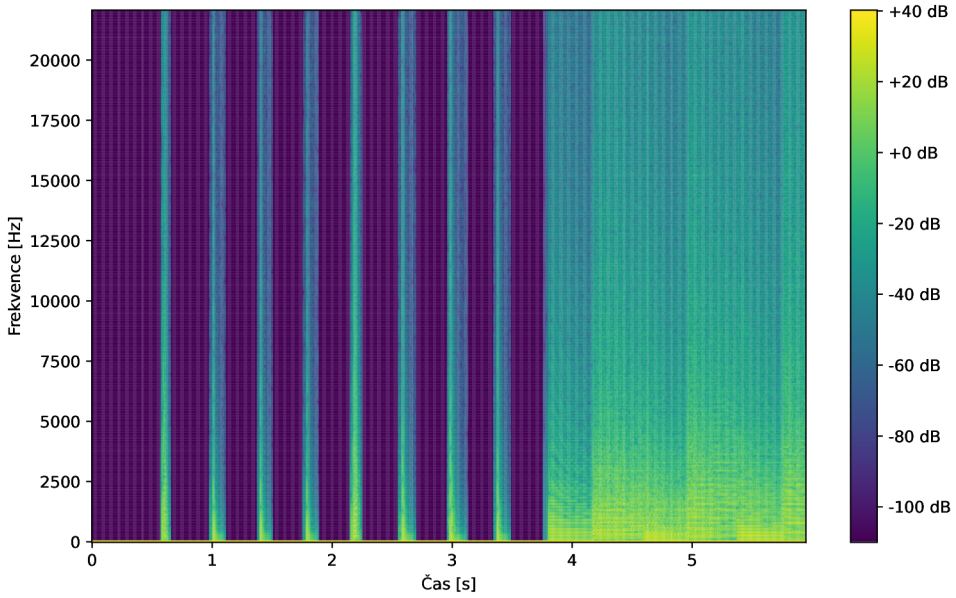
Pro načítání signálu z datasetu byl vytvořen soubor `dataset.py`, v němž jsou definovány parametry zvukových souborů, které mají být pro další práci s modelem načteny. Tento soubor zároveň umožňuje rozdělit si soubory v datasetu tak, aby jedna část souborů byla použita pouze pro trénování modelu, a druhá část pouze pro testování. Soubory jsou vzorkovány s frekvencí 44,1 kHz. Signál je načítán po deseti segmentech, přičemž každý z nich je o velikosti 262 144 vzorků, což odpovídá délce cca 5,94 s. Načítaný signál je před dalším zpracováním normalizován tak, aby jeho úroveň nepřekročila meze intervalu $(-1, 1)$.

4.1.2 Úprava vstupních dat

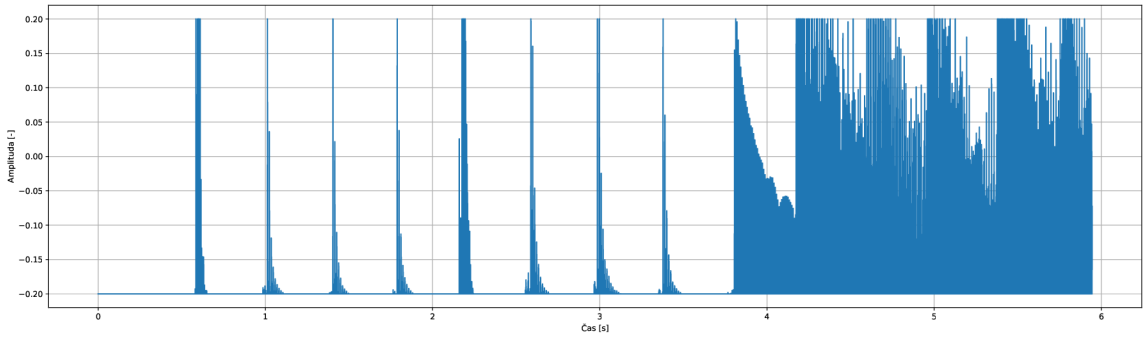
Před samotným procesem vzorkování je vstupní signál nezbytné upravit několika způsoby. Prvním způsobem úpravy je maximální zašumění signálu tak, abychom zís-

kali signál v podobě x_T, z něhož vycházíme v rámci reverzního difuzního procesu, tedy vzorkování z Gaussovského šumu.

V rámci druhého způsobu úpravy je ve smyčce získáván podmiňovací signál y umělým zkreslením čistého vstupního signálu clippingem s deseti různými úrovněmi prahu zkreslení (thresholdu), a sice od úrovně 0,05 po 0,5 s krokem 0,05. Vzorkování tedy probíhá pro každý vstupní signál desetkrát, tedy pro každou prahovou úroveň samostatně. Níže je pro ilustraci přiložen spektrogram 4.1 a graf časového průběhu 4.2 pro signál zkreslený clippingem s prahovou úrovní 0,2.



Obr. 4.1: Spektrogram signálu uměle zkresleného clippingem s thresholdem 0,2.



Obr. 4.2: Časový průběh signálu uměle zkresleného clippingem s thresholdem 0,2.

4.1.3 Difuzní čas a šumový rozptyl

Pomocí funkce `get_time_schedule` je dle [30] vypočten difuzní čas τ v podobě konečného počtu diskretizovaných kroků na základě rovnice:

$$\tau_i = \left(\sigma_{max}{}^{\frac{1}{\rho}} + \frac{i}{\mathrm{T}-1}\left(\sigma_{min}{}^{\frac{1}{\rho}} - \sigma_{max}{}^{\frac{1}{\rho}}\right)\right)^{\rho}, \tag{4.1}$$

kde platí $0 \leq i < \mathrm{T}$, přičemž T je počet kroků. Dále σ_{min} a σ_{max} jsou krajní hodnoty úrovně šumu, parametr ρ slouží k úpravě rozprostření počtu kroků τ napříč úrovněmi šumu $\sigma(\tau)$. V rámci našeho algoritmu je dle doporučení [30] $\sigma(\tau) = \tau$, ostatní z výše uvedených parametrů jsou aplikovány s následujícími hodnotami: $\mathrm{T} = 100$, $\sigma_{min} = 10^{-6}$, $\sigma_{max} = 10$ a $\rho = 7$.

4.1.4 Deterministický sampler

Sampler, vzorkující signál z Gaussovského šumu, byl navržen pro již natrénovaný U-net model neuronové sítě, poskytnutý společně s váhami vedoucím této bakalářské práce [31]. Model i váhy se načítají pomocí knihovny Hydra [32].

V kapitole 2 jsme se věnovali popisu reverzního difuzního procesu především pomocí stochastických diferenčních rovnic. V praktické části této bakalářské práce jsme však algoritmus pro aposteriorní vzorkování realizovali v podobě deterministického sampleru, jehož funkce je popsána obyčejnými diferenciálními rovnicemi.

Nejdůležitější částí algoritmu je vzorkovací smyčka, která začíná předzpracováním signálu dle [30]. V této fázi jsou pomocí funkce `get_preconditioning` nejprve vypočteny hodnoty škálovacích parametrů $\mathrm{c_{skip}}(\tau)$, $\mathrm{c_{out}}(\tau)$, $\mathrm{c_{in}}(\tau)$ a $\mathrm{c_{noise}}(\tau)$, které jsou nezbytné pro zjištění odhadu $\hat{\mathrm{x}}_0$ výpočtem:

$$D_\theta(\mathrm{x}_\tau, \tau) = \mathrm{c_{skip}}(\tau)\mathrm{x}_\tau + \mathrm{c_{out}}(\tau)F_\theta\Big(\mathrm{c_{in}}(\tau)\mathrm{x}_\tau, \mathrm{c_{noise}}(\tau)\Big), \tag{4.2}$$

kde D_θ je denoiser, x_τ je náš zašuměný signál a F_θ je U-net model, aplikující CQT. Na takto získaný odhad $\hat{\mathrm{x}}_0$ je následně aplikován filtr typu horní prospust, jehož účelem je potlačit frekvence, které CQT nezpracuje. [31]

Následuje výpočet funkce skóre $\nabla_{\mathrm{x}} \log p_\tau(\mathrm{x}_\tau)$ na základě vztahu 2.6, kde odhad střední hodnoty $\mu(\mathrm{x}_0, \tau)$ je naše $\hat{\mathrm{x}}_0$. Dále je realizován výpočet diferenční rovnice pravděpodobnostního toku (Probability Flow ODE) dle [31], která je ve své podstatě deterministickou částí rovnice 2.5:

$$\mathrm{dx} = -\tau\nabla_{\mathrm{x}} \log p_\tau(\mathrm{x}_\tau)\mathrm{d}\tau. \tag{4.3}$$

Pro celý proces vzorkování je velmi důležitá funkce `get_likelihood_score`, navržena dle [1], která realizuje výpočet gradientu:

$$\nabla_{\mathrm{x}_\tau} \log p(\mathrm{y}|\mathrm{x}_\tau) \simeq -\zeta(\tau)\nabla_{\mathrm{x}_\tau}\|\mathrm{y} - \mathcal{A}(D_\theta(\mathrm{x}_\tau, \tau))\|_2^2. \tag{4.4}$$

Tento element z celého generativního procesu činí proces podmíněný, neboť vychyluje vzorkování směrem, který je co nejbližší pravděpodobnému výsledku. Zmíněnou metodu lze označit za vedení rekonstrukce (reconstruction guidance) [1]. Operátor poškození $\mathcal{A}$ zde značí zkreslení signálu clippingem stejným způsobem, jako při úpravě vstupního signálu do podoby y. Parametr $\zeta(\tau)$ je škálovací funkce, která udává, do jaké míry je vzorkovací proces ovlivněn podmíněním. Při realizaci našeho algoritmu byla zvolena hodnota $\zeta(\tau) = 0,2$.

Na konci každé jednotlivé smyčky získáme $\mathrm{x}_{\tau_{i-1}}$, tedy stav x_τ, z něhož vycházíme v následující iteraci vzorkovací smyčky [20]. Výstupem algoritmu po průběhu poslední iterace je rekonstruovaný signál.

Algorithm 1 Vzorkovací smyčka

Require: σ_{max}, počet kroků T, $\sigma(\tau_i) = \tau_i$, U-net model F_θ, zkreslený signál y

$\mathrm{x_T} \sim \mathcal{N}(\mathbf{0}, \sigma_{max}^2\mathbf{I})$ ▷ Zašuměný signál
for $i = \mathrm{T}, \ldots, 1$ **do** ▷ Diskretizované kroky
  $\mathrm{c_{skip}}(\tau_\mathrm{i}), \mathrm{c_{out}}(\tau_i), \mathrm{c_{in}}(\tau_i), \mathrm{c_{noise}}(\tau_i) \leftarrow \mathrm{preconditioning}(\tau_i)$
  $\hat{\mathrm{x}}_0 = \mathrm{c_{skip}}(\tau_i)\mathrm{x}_{\tau_i} + \mathrm{c_{out}}(\tau_i)F_\theta\Big(\mathrm{c_{in}}(\tau_i)\mathrm{x}_{\tau_i}, \mathrm{c_{noise}}(\tau_i)\Big)$ ▷ Předzpracování
  $\hat{\mathrm{x}}_0 = H_{hpf}(\hat{\mathrm{x}}_0)$ ▷ Filtr (horní propust)
  $s_\theta = (\hat{\mathrm{x}}_0 - \mathrm{x}_{\tau_i})/{\tau_i}^2$ ▷ Funkce skóre
  $\mathrm{dx} = -\tau_i(s_\theta - \zeta(\tau_i)\nabla_{\mathrm{x}_\tau}\|\mathrm{y} - \mathcal{A}(D_\theta(\mathrm{x}_{\tau_i}, \tau_i))\|_2^2)$ ▷ Podmíněná rekonstrukce
  $\mathrm{x}_{\tau_{i-1}} = \mathrm{x}_{\tau_i} + (\tau_{i+1} - \tau_i)\mathrm{dx}$ ▷ Stav v následujícím kroku
end for
return $\hat{\mathrm{x}}_0$ ▷ Rekonstruovaný signál

4.2 Hodnocení kvality rekonstrukce

Na základě deseti vstupních signálů byl pomocí deterministického sampleru signál rekonstruován deseti možnými způsoby (tj. procesem, podmíněným na základě deseti různých variant uměle poškozeného signálu y). Posuzováno tedy bylo celkem 100 verzí rekonstruovaného signálu. Pro zhodnocení kvality rekonstrukce byly zvoleny následující objektivní metriky: Diff-SDR, ℓ_1 norma a ℓ_2 norma.

4.2.1 Poměr signálu ke zkreslení

Metrika Signal to Distortion Ratio (SDR), neboli poměr energie signálu k energii zkreslení, je metoda, posuzující vzájemnou podobu čistého vstupního signálu se signálem rekonstruovaným z hlediska amplitudy a času na základě vztahu:

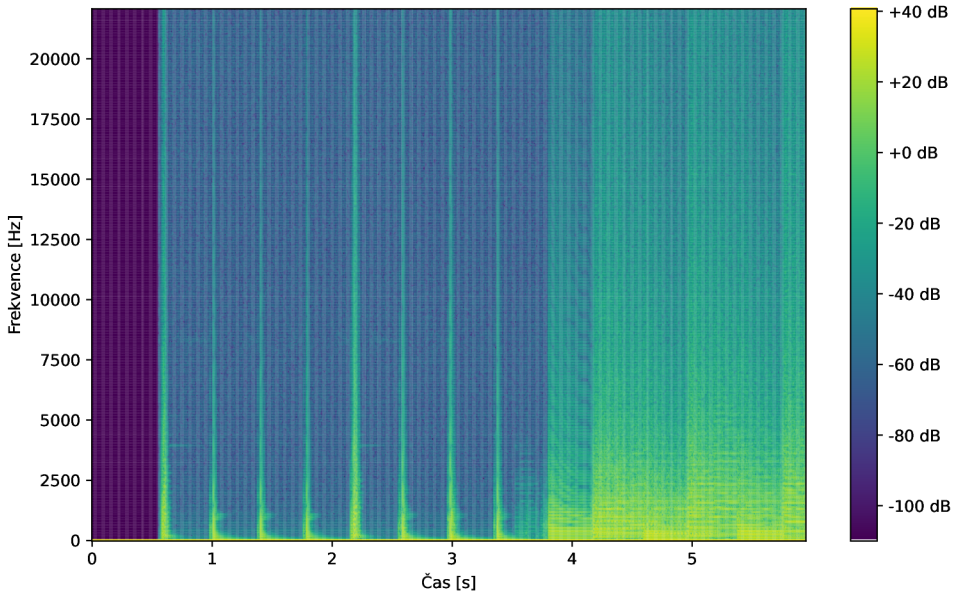
$$\mathrm{SDR}(\mathrm{x}_0, \hat{\mathrm{x}}_0) = 10 \cdot \log_{10} \frac{\|\mathrm{x}_0\|_2^2}{\|\mathrm{x}_0 - \hat{\mathrm{x}}_0\|_2^2} \quad [\mathrm{dB}], \tag{4.5}$$

kde x_0 je původní signál a $\hat{\mathrm{x}}_0$ signál rekonstruovaný [33]. Tuto metriku však k zhodnocení kvality výstupu našeho algoritmu využijeme v podobě:

$$\text{Diff-SDR} = \mathrm{SDR}(\mathrm{x}_0, \hat{\mathrm{x}}_0) - \mathrm{SDR}(\mathrm{x}_0, \mathrm{y}), \tag{4.6}$$

která umožňuje zjistit, nakolik se hodnota SDR zlepšila pro rekonstruovaný signál $\hat{\mathrm{x}}_0$ oproti SDR poškozeného signálu y. Obecně platí, že vysoká úroveň Diff-SDR značí vysokou kvalitu rekonstrukce signálu, je tedy žádoucí dosáhnout úrovně co nejvyšší.

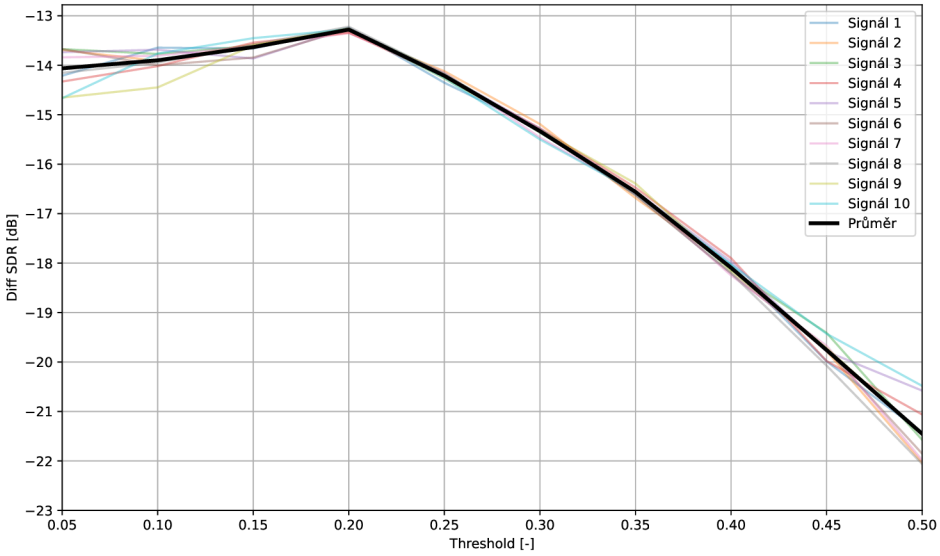
Dle průběhu hodnot SDR v závislosti na úrovni poškození signálu lze vyvodit, že pro signál s výraznějším poškozením (tedy pro hodnoty thresholdu nižší) bude zlepšení SDR výraznější. Naproti tomu málo poškozený signál je méně obtížné rekonstruovat, proto rozdíl v SDR poškozeného a rekonstruovaného signálu bude nízký. Proč je zlomový bod v grafu 4.4 právě na úrovni thresholdu 0,2, může napovědět porovnání spektrogramu 4.1, který vyobrazuje signál s clippingem právě na této úrovni, vůči spektrogramu 4.3, kde je signál zkreslený s thresholdem 0,25.



Obr. 4.3: Spektrogram signálu uměle zkresleného clippingem s thresholdem 0,25.

4.2.2 Střední absolutní chyba

Střední absolutní chyba, neboli ℓ_1 norma, patří společně s ℓ_2 normou k hojně využívaným metrikám pro posuzování audio signálů. Výpočet ℓ_1 normy spočívá ve výpočtu rozdílů jednotlivých bodů průběhu signálu v čase a následným zprůměrová-



Obr. 4.4: Evaluace rekonstruovaného signálu pomocí Diff SDR.

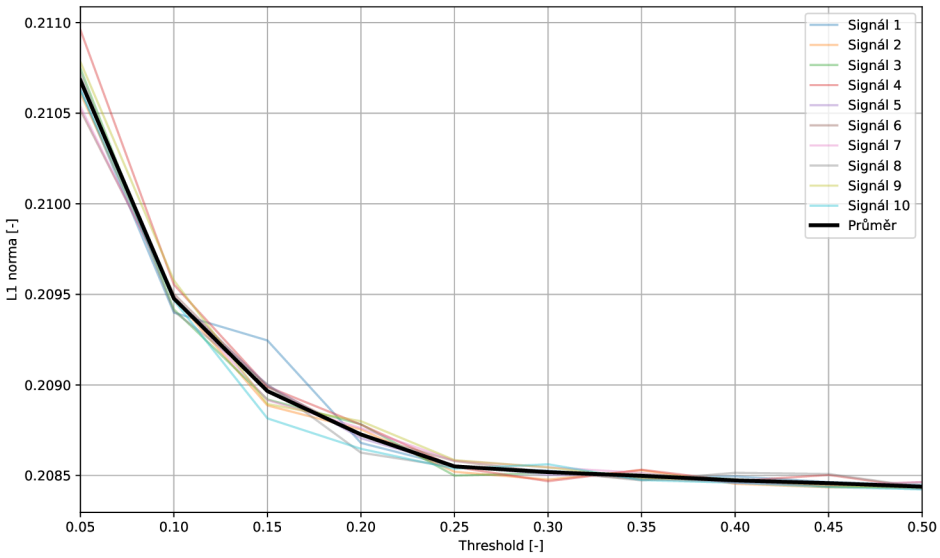
ním absolutních hodnot těchto rozdílů [33]. Všechny odchylky jsou touto normou penalizovány se stejnou citlivostí. V případě obou metod je žádoucí, abychom dosáhli co nejnižších hodnot chyby, neboť to značí větší vzájemnou podobnost obou porovnávaných signálů.

Budeme-li touto metodou hodnotit odchylku rekonstruovaného signálu $\hat{x}_0$ od původního nepoškozeného signálu x_0, lze očekávat, že v případě rekonstrukce silněji poškozeného signálu budou odchylky četnější než pro signál méně poškozený. Vzhledem k tomu, že nižší hodnota normy ℓ_1 značí nižší výskyt odchylek, lze říci, že její průběh v grafu 4.5 je v souladu s očekáváním, neboť dochází k jejímu poklesu směrem k vyšším hodnotám thresholdu, které znamenají nižší úroveň poškození signálu.

4.2.3 Střední kvadratická chyba

Střední kvadratická chyba, neboli ℓ_2 norma slouží, stejně jako ℓ_1 norma, pro posouzení odchylek průběhu signálu v čase, ovšem s tím rozdílem, že větší odchylky penalizuje výrazněji než odchylky menší [33]. Podobně jako v případě normy ℓ_1 je i zde žádoucí dosáhnout co nejnižší hodnoty chyby.

Pro evaluaci rekonstruovaného signálu byla využita metoda ℓ_2 komprimované STFT. Tato funkce byla aplikována i v rámci našeho algoritmu deterministického sampleru, konkrétně při výpočtu pravděpodobnostního skóre 4.4. Její princip spočívá v tom, že nejprve oba signály (tedy původní x_0 a rekonstruované $\hat{x}_0$) podrobí krátkodobé Fourierově transformaci (STFT), a následně jsou obě magnitudová spektra komprimována a vzájemně porovnána pomocí ℓ_2 normy.



Obr. 4.5: Evaluace rekonstruovaného signálu pomocí ℓ_1 normy.

Výpočet probíhá dle vztahu definovaného v [31] následujícím způsobem:

$$C(\mathbf{y}, \hat{\mathbf{y}}) = \frac{1}{M} \sum_{m=1}^{M} \sum_{k=1}^{K} |S_{\text{comp}}(\mathbf{y})_{m,k} - S_{\text{comp}}(\hat{\mathbf{y}})_{m,k}|^2, \tag{4.7}$$

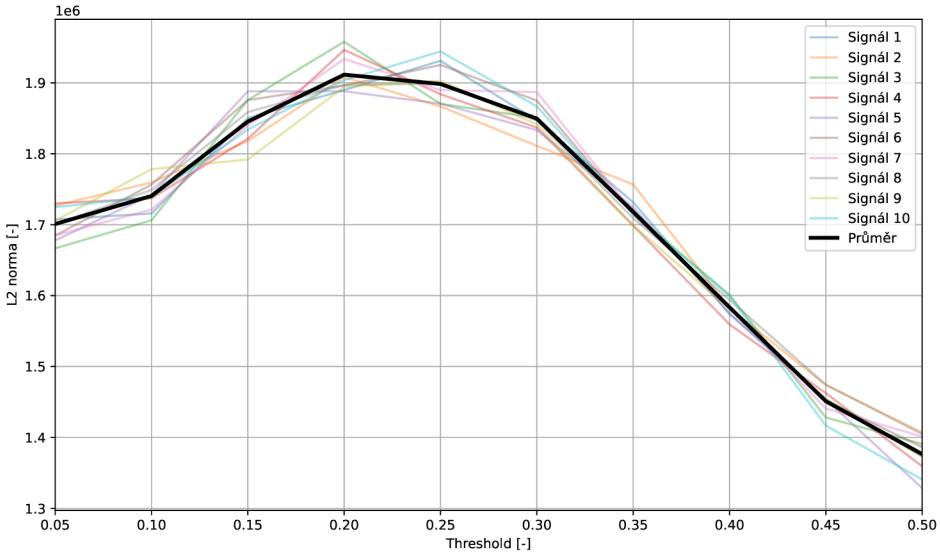
kde signál $\hat{\mathrm{y}}$ je odhad $\hat{\mathrm{x}}_0$, poškozený známým operátorem $\mathcal{A}$ (viz. 7. řádek algoritmu 1), $S_{\text{comp}}(\mathbf{y}) = |\text{STFT}(\mathbf{y})|^{2/3} \exp j\angle\text{STFT}(\mathbf{y})$ je komprimovaný STFT spektrogram, parametr M je počet časových oken a K značí počet frekvenčních složek.

Jak lze opět snadno předpovědět, nejnižších odchylek od vstupního signálu, dosahují po rekonstrukci signály při nejnižší úrovni poškození. Je naopak překvapivé, že nejvyšší hodnoty nezaznamenáváme u nejsilněji poškozeného signálu, nýbrž u signálů, zkreslených clippingem s thresholdem 0,15 až 0,30.

4.2.4 Porovnání výstupů použitých metrik

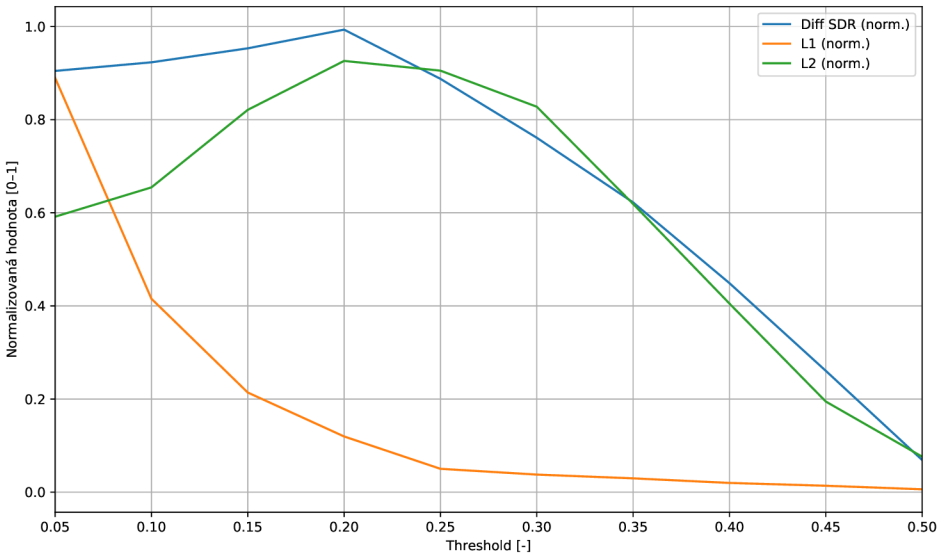
Pro porovnání výstupů jednotlivých metrik byly zprůměrované hodnoty každé z nich normalizovány a následně vyneseny do společného grafu 4.7. Lze pozorovat, že s rostoucí hodnotou thresholdu, tedy s nižší úrovní poškození signálu hodnoty všech metrik klesají. Je pochopitelné, že menší míru poškození je snazší zvrátit s vyšší přesností, neboť podpůrných dat pro rekonstrukci je v signálu zachováno více. Rozdíl v míře poškození signálu je nejvýraznější mezi hodnotami thresholdu 0,2 a 0,25 (viz. spektrogramy 4.1 a 4.3), což je při evaluaci vhodné brát v potaz.

Průběh Diff-SDR značí, že pro signály poškozené s thresholdem 0,2 a nižším došlo k nejvýraznějšímu zlepšení SDR po jejich rekonstrukci. Nad touto hodnotou již rozdíl SDR klesá, ale není nulový, poškození signálu se zde stále projevuje.



Obr. 4.6: Evaluace rekonstruovaného signálu pomocí ℓ_2 normy.

V případě ℓ_1 normy a ℓ_2 normy vidíme, že odchylky rekonstruovaného signálu $\hat{x}_0$ od referenčního signálu x_0 jsou pro vyšší úroveň thresholdu méně výrazné, a vyskytují se s nižší četností. Naopak po rekonstrukci signálu s nejsilnějším poškozením dosahuje ℓ_1 norma nejvyšších hodnot, zároveň zde převyšuje hodnotu ℓ_2 normy, což znamená, že odchylky rekonstruovaného signálu byly sice četné, ale nebyly tolik výrazné jako v případě signálů s poškozením na úrovni thresholdu 0,15 až 0,30. Pro hodnoty thresholdu nad touto úrovní již ℓ_2 norma klesá, což značí, že se snižuje také míra odchylek.



Obr. 4.7: Porovnání výstupů použitých metrik.

Závěr

Tato bakalářská práce se zabývá využitím difuzních modelů neuronových sítí při řešení inverzních úloh, konkrétně realizací vzorkování z aposteriorního pravděpodobnostního rozdělení. Cílem práce bylo implementovat řešení dle [1] a poté navrhnout modifikaci pro vylepšení této metody, případně pro její využití v jiné rekonstrukční úloze (audio inpainting, zvyšování šířky pásma, atd.).

Problematika byla nejprve popsána v rámci teoretické části od základů neuronových sítí přes princip difuzních modelů až po jejich roli v řešení inverzních úloh prostřednictvím difuzního aposteriorního vzorkování. Představili jsme si také různé typy inverzních úloh, na které je možné se zaměřit.

Na základě předem natrénovaného U-net modelu difuzní neuronové sítě, poskytnutého vedoucím této bakalářské práce [31], byl v jazyku Python vytvořen funkční algoritmus determinstického sampleru, který je schopen rekonstruovat audio signál podmíněným procesem vzorkování z Gaussovského šumu. Testování algoritmu proběhlo na deseti vstupních signálech, které byly uměle poškozeny clippingem s deseti různými úrovněmi thresholdu od 0,05 do 0,5. Pro tento účel jsme využili nahrávky kytary z datasetu IDMT-SMT-GUITAR [29].

Původním záměrem bylo realizovat sampler stochastický, vzhledem k náročnosti provedení jsme se však namísto toho zaměřili na variantu sampleru deterministického. Výsledkem je funkční algoritmus schopný declippingu poškozeného signálu.

Kvalita rekonstruovaného signálu byla zhodnocena v závislosti na úrovni jeho poškození za pomoci objektivních metrik, a sice Diff-SDR, ℓ_1 normy a ℓ_2 normy. Z výstupů dle očekávání vyplývá, že kvalita rekonstrukce je nejvyšší v případě nejméně poškozeného signálu. Co se týče kvality rekonstrukce silněji poškozeného signálu je zde nepochybně prostor pro zlepšení, i zde jsou však výsledky obstojné.

Vzhledem k náročnosti úlohy se již bohužel nepodařilo metodu dále modifikovat, nabízí se zde však možnosti úpravy algoritmu například využitím binární masky pro realizaci inpaintingu. Další variantou je použít již vytvořený algoritmus deterministického sampleru jako základ pro dokončení původně zamýšleného provedení sampleru stochastického.

Literatura

[1] MOLINER, Eloi; LEHTINEN, Jaakko; VÄLIMÄKI, Vesa. *Solving Audio Inverse Problems with a Diffusion Model.* Online. ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Rhodes Island, Řecko. 2023. Dostupné z: `https://ieeexplore.ieee.org/document/10095637/authors` [cit. 2024-10-10].

[2] PYTORCH. Online. 2025. Dostupné z: `https://pytorch.org/`. [cit. 2025-06-02].

[3] HARDESTY, Larry. *Explained: Neural networks.* Online. Massachusetts Institute of Technology. 2017. Dostupné z: `https://news.mit.edu/2017/explained-neural-networks-deep-learning-0414`. [cit. 2024-11-10].

[4] CHOLLET, François. *Deep learning v jazyku Python: Knihovny Keras, TensorFlow.* Praha: Grada Publishing. 2019. ISBN 978-80-271-2750-4.

[5] INNOKREA. *Machine Data Learning - Part 3: Data Representation Methods.* Online. 2023. Dostupné z: `https://www.innokrea.com/machine-learning-part-3-data-representation-methods/`. [cit. 2024-11-30]

[6] SHKULOV, Valentine. *Neural Network Layers: All You Need Is Inside Comprehensive Overview.* Online. 2023. Dostupné z: `https://hackernoon.com/neural-network-layers-all-you-need-is-an-inside-comprehensive-overview/`. [cit. 2024-11-30].

[7] PYTORCH 2.5 DOCUMENTATION. *Linear.* Online. 2023. Dostupné z: `https://pytorch.org/docs/stable/generated/torch.nn.Linear.html`. [cit. 2024-12-04].

[8] AIM (ANALYTICS INDIA MAG). *Dense Layers.* Online. 2024. Dostupné z: `https://analyticsindiamag.com/topics/what-is-dense-layer-in-neural-network/`. [cit. 2024-12-05].

[9] GEEKSFORGEEKS. *Activation functions in Neural Networks.* Online. 2024. Dostupné z: `https://www.geeksforgeeks.org/activation-functions-neural-networks/`. [cit. 2024-12-05].

[10] BROWNLEE, Jason. *What Is a Gradient in Machine Learning?* Online. Machine Learning Mastery. 2021. Dostupné z: `https://machinelearningmastery.com/gradient-in-machine-learning/`. [cit. 2024-12-05].

[11] WIKISKRIPTA. *Gradient.* Online. 2021. Dostupné z: `https://www.wikiskripta.eu/index.php?title=Gradient&oldid=470966`. [cit. 2024-12-05].

[12] SUBRAMANIAN, Suraj; JUAREZ, Seth; BREVIU, Cassie; SOSHNIKOV, Dmitry a BORNSTEIN, Ari. *Introduction to PyTorch: Optimizing Model Parameters*. Online. Dostupné z: https://pytorch.org/tutorials/beginner/basics/optimization_tutorial.html. [cit. 2024-12-1].

[13] NORDIC TELECOM. *Neuronové sítě: Jak umělá inteligence napodobuje lidský mozek*. Online. 2024. Dostupné z: https://nordictelecom.cz/novinky/105/neuronova-sit/#h5. [cit. 2024-12-1].

[14] JAISWAL, Abishek. *Understanding Optimizers in Neural Networks*. Online. Medium. 2023. Dostupné z:https://medium.com/@abhishek.jaiswaal1810/optimizers-in-neural-networks-4fb7adee4a63. [cit. 2024-12-05].

[15] WENG, Lilian. *What are Diffusion Models?* Online. Lil'Log. 2021. Dostupné z: https://lilianweng.github.io/posts/2021-07-11-diffusion-models/. [cit. 2024-11-15].

[16] YANG Ling; ZHANG Zhilong; SONG Yang; HONG Shenda; XU Runsheng et al. *Diffusion Models: A Comprehensive Survey of Methods and Applications*. Online. 2022. Dostupné z: https://arxiv.org/abs/2209.00796. [cit. 2024-11-15].

[17] DALL · E 2. Online. OpenAI. Dostupné z: https://openai.com/index/dall-e-2/. [cit. 2025-06-02].

[18] O'CONNOR, Ryan. *Introduction to Diffusion Models for Machine Learning*. Online. AssemblyAI. 2022. Dostupné z: https://www.assemblyai.com/blog/diffusion-models-for-machine-learning-introduction/. [cit. 2024-11-15].

[19] HRON, Karel; FAČEVICOVÁ, Kamila a KUNDEROVÁ, Pavla. *Markovovy řetězce a jejich aplikace*. Univerzita Palackého v Olomouci. 2018. ISBN 978-80-244-5432-0. Dostupné také z: https://www.bookport.cz/kniha/markovovy-retezce-a-jejich-aplikace-14118/. [cit. 2024-11-26].

[20] LEMERCIER, Jean-Marie; RICHTER, Julius; WELKER, Simon; MOLINER, Eloi; VÄLIMÄKI, Vesa, et. al. *Diffusion Models for Audio Restoration*. Online. 2024. Dostupné z: https://arxiv.org/abs/2402.09821 [cit. 2024-10-22].

[21] TME ELECTRONIC COMPONENTS. *Digitální a analogové signály a průběhy*. Online. 2024. Dostupné z: https://www.tme.eu/cz/news/library-articles/page/61486/digitalni-a-analogove-signaly-a-prubehy/. [cit. 2024-12-04].

[22] LU, Ye-Xin; AI, Yang; DU, Hui-Peng; LING, Zhen-Hua. *Towards High-Quality and Efficient Speech Bandwidth Extension with Parallel Amplitude and Phase Prediction* Online. 2024. Dostupné z: `https://arxiv.org/html/2401.06387v1`. [cit. 2024-12-04].

[23] MOLINER, Eloi; VÄLIMÄKI, Vesa. *Diffusion-Based Audio Inpainting.* Online. Journal of the Audio Engineering Society. 2023. Dostupné z: `https://arxiv.org/abs/2305.15266`. [cit. 2024-10-10].

[24] BRUNNOTTS, Kate. *Clipping in Audio: The Complete Beginner's Guide.* Online. EMastered Blog. 2022. Dostupné z: `https://emastered.com/blog/what-is-clipping.`. [cit. 2024-12-05].

[25] ZÁVIŠKA, Pavel; RAJMIC, Pavel; OZEROV, Alexey a RENCKER, Lucas. *A survey and an extensive evaluation of popular audio declipping methods.* Online. IEEE Journal of Selected Topics in Signal Processing. 2020, roč. 15, č. 1, s. 5-24. ISSN 1941-0484. Dostupné z: `https://doi.org/10.1109/JSTSP.2020.3042071`. [cit. 2024-12-05].

[26] MIKULÁŠKOVÁ, Aneta. *Moderní metody rekonstrukce saturovaných audio signálů.*. Brno: Brno University of Technology, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací, 2022, 59 p. Diplomová práce. Vedoucí práce: prof. Mgr. Pavel Rajmic, Ph.D. Dostupné z: `https://www.vut.cz/studenti/zav-prace/detail/145352`. [cit. 2025-06-2].

[27] CHUNG, Hyungjin; KIM, Jeongsol; MCCANN, Michael T.; KLASKY, Marc L.; YE, Jongchul. *Diffusion Posterior Sampling for General Noisy Inverse Problems.* Online. The Eleventh International Conference on Learning Representations. 2023. Dostupné z: `https://openreview.net/forum?id=OnD9zGAGT0k`. [cit. 2025-05-05].

[28] SCHNEPPAT, Jörg-Owe; *Bayesian Inference and Posterior Distributions.* Online. Schneppat AI. Dostupné z: `https://schneppat.com/bayesian-inference-and-posterior-distributions.html`. [cit. 2025-05-11].

[29] KEHLING, Christian; ABESSER, Jakob; DITTMAR, Christian; SCHULLER, Gerald. *Automatic Tablature Transcription of Electric Guitar Recordings by Estimation of Score- and Instrument-related Parameters.* Online. Proceedings of the 17th International Conference on Digital Audio Effects. Erlangen, Germany. 2014. Dostupné z: `https://www.idmt.fraunhofer.de/en/publications/datasets/guitar.html`. [cit. 2025-05-25].

[30] KARRAS, Tero; AITTALA, Miika; AILA, Timo; LAINE, Samuli. *Elucidating the Design Space of Diffusion-Based Generative Models.* Online. 2022. Dostupné z: https://arxiv.org/abs/2206.00364. [cit. 2025-06-01].

[31] ŠVENTO, Michal; MOLINER, Eloi; JUVELA, Lauri; WRIGHT, Alec; VÄLIMÄKI, Vesa. *Estimation and Restoration of Unknown Nonlinear Distortion using Diffusion.* Online. Journal of Audio Engineering Society. 2025. Dostupné z: https://arxiv.org/abs/2501.05959. [cit. 2025-05-29].

[32] YADAN, Omry. *Hydra – A framework for elegantly configuring complex applications.* Online. 2019. Dostupné z: https://github.com/facebookresearch/hydra. [cit. 2025-05-30].

[33] RADKOFF, Eva. *Loss Functions in Audio ML.* Online. Sounds and Words. 2021. Dostupné z: https://www.soundsandwords.io/audio-loss-functions/. [cit. 2025-05-30].

Seznam symbolů a zkratek

CQT transformace s konstantním Q – Constant-Q Transform

LSTM vrstvy s dlouhou krátkodobou pamětí – Long Short-Term Memory

MAE střední absolutní chyba – Mean Absolute Error

MSE střední kvadratická chyba – Mean Squared Error

SDR poměr signálu ke zkreslení – Signal to Distortion Ratio

SGD stochastický gradientní sestup – Stochastic Gradient Descent

STFT krátkodobá Fourierova transformace – Short Time Fourier Transform

5 Obsah elektronické přílohy

Níže je uveden seznam a stručný popis souborů obsažených v elektronické příloze. Pro vytvoření kódu byl použit Python 3.10.11 a PyTorch 2.6.0. Pro spuštění algoritmu slouží soubor `sampler_det.py`, v němž je definovaná hlavní část kódu. Obsah této přílohy je rovněž přístupný ve veřejném GitHub repozitáři na adrese `https://github.com/247800/DPSampler/tree/master`, kde je zároveň k dispozici soubor `README.md` s instrukcemi pro instalaci modelu, vah a dalších souborů, nezbytných pro správnou funkčnost algoritmu.

```
DPSampler.zip./ ........................... kořenový adresář přiloženého archivu
├── utils
│   ├── callback.py ........ funkce pro export grafů, spektrogramů a audio souborů
│   ├── loss.py ............................................ ztrátová funkce
│   └── sampling_utils.py ......................... pomocné funkce pro sampler
├── dataset.py ..................................... skript pro načtení datasetu
├── sampler_det.py ......................... hlavní skript pro spuštění algoritmu
└── requirements.txt ................................... výpis použitých balíčků
```