دانشگاه صنعتی امیرکبیر
(پلی‌تکنیک تهران)
دانشکده مهندسی کامپیوتر

پایان‌نامه کارشناسی‌ارشد

گرایش هوش مصنوعی و رباتیکز

بهبود سرعت مدل‌های پخشی

نگارش

عطیه غفارلوی مقدم

استاد راهنما

دکتر احمد نیک‌آبادی

دی‌ماه ۱۴۰۳

بسم الله الرحمن الرحیم

به نام خدا

تعهدنامه اصالت اثر

تاریخ: دی‌ماه ۱۴۰۳

دانشگاه صنعتی امیرکبیر
(پلی‌تکنیک تهران)

اینجانب عطیه غفارلوی مقدم متعهد می‌شوم که مطالب مندرج در این پایان‌نامه حاصل کار پژوهشی اینجانب تحت نظارت و راهنمایی اساتید دانشگاه صنعتی امیرکبیر بوده و به دستاوردهای دیگران که در این پژوهش از آنها استفاده شده است مطابق مقررات و روال متعارف ارجاع و در فهرست منابع و مآخذ ذکر گردیده است. این پایان‌نامه قبلاً برای احراز هیچ مدرک هم‌سطح یا بالاتر ارائه نگردیده است.

در صورت اثبات تخلف در هر زمان، مدرک تحصیلی صادر شده توسط دانشگاه از درجه اعتبار ساقط بوده و دانشگاه حق پیگیری قانونی خواهد داشت.

کلیه نتایج و حقوق حاصل از این پایان‌نامه متعلق به دانشگاه صنعتی امیرکبیر می‌باشد. هرگونه استفاده از نتایج علمی و عملی، واگذاری اطلاعات به دیگران یا چاپ و تکثیر، نسخه‌برداری، ترجمه و اقتباس از این پایان نامه بدون موافقت کتبی دانشگاه صنعتی امیرکبیر ممنوع است. نقل مطالب با ذکر مآخذ بلامانع است.

عطیه غفارلوی مقدم

امضا

چکیده

مدل‌های پخشی به عنوان یک خانواده‌ی قدرتمند از مدل‌های مولد با نتایج قابل توجه در حوزه‌های مختلف تصویر، صوت، متن و گراف، در سال‌های اخیر مورد توجه بسیاری از محققین یادگیری ماشین قرار گرفته‌اند. همه‌ی مدل‌های مطرح شده در این خانواده از یک رویکرد مشابه پیروی می‌کنند و آن هم تخریب داده با نویز در یک فرآیند گام به گام و سپس بازگرداندن این فرآیند برای تولید نمونه‌های جدید از داده می‌باشد. مشکل اصلی این دسته از مدل‌های مولد سرعت کم فرآیند تولید نمونه‌ی آنها است که باعث می‌شود علیرغم کیفیت و تنوع نمونه‌های تولیدی، در بسیاری از کاربردهای بلادرنگ و تعاملی قابل استفاده نباشند. روش‌های افزایش سرعت مدل‌های پخشی بر دو قِسم روش‌های کاهش تعداد گام و روش‌های کاهش محاسبات در هر گام است. در روش‌های کاهش محاسبات در هر گام از فشرده‌سازی شبکه یا استفاده از حافظه‌ی پنهان برای ذخیره‌ی بخشی از محاسبات استفاده می‌شود. یک رویکرد رایج در همین دسته، رویکرد ادغام توکن‌هاست که با درنظر گرفتن هزینه‌ی محاسباتی درجه‌ی دوی مبدل‌ها نسبت به اندازه‌ی ورودی سعی در کاهش اندازه‌ی ورودی برای کاهش حجم محاسبات در این بلوک‌ها را دارد. این روش علیرغم مزایای آن نقاط ضعفی از جمله ثابت گرفتن درصد ادغام توکن‌ها و محاسبه‌ی ماتریس مشابهت توکن‌ها، که خود سربار زمانی اضافه می‌کند، را دارد. در این پژوهش تلاش می‌شود تا فرآیند ادغام با تعریف یک آستانه به صورت تطبیق‌پذیر انجام شود و برای کاهش سربار محاسباتی ناشی از محاسبه‌ی ماتریس مشابهت توکن‌ها نیز از یک حافظه‌ی پنهان برای ذخیره و استفاده‌ی مجدد از محاسبات استفاده شود. روش پیشنهادی نسبت به مدل پایه‌ی پخشی افزایش سرعتی با ضریب ۱.۲۴ دارد که می‌توان نشان داد بیشترین کاهشی است که در روش ادغام توکن قابل دستیابی است و این امر بدون کاهش کیفیت تصاویر تولیدی مدل پخشی انجام شده است.

واژه‌های کلیدی:

مدل‌های مولد، مدل‌های پخشی، مدل‌های پخشی نویززدای احتمالاتی، ادغام توکن‌ها، استفاده از حافظه‌ی پنهان

فهرست مطالب

عنوان | صفحه

۱ مقدمه ۱

۲ مدل‌های پخشی و رویکردهای افزایش سرعت ۴
۲-۱ تعاریف ۵
۲-۱-۱ مدل‌های مولد ۵
۲-۱-۲ مدل‌های پخشی ۵
۲-۱-۳ مدل‌های پخشی نویززدای احتمالاتی ۷
۲-۱-۳-۱ تعریف مدل ۷
۲-۱-۳-۲ مسیر رو به جلو ۷
۲-۱-۳-۳ مسیر رو به عقب ۸
۲-۱-۳-۴ تعریف تابع خطا ۹
۲-۱-۳-۵ تغییر پارامتر تابع خطا ۹
۲-۱-۳-۶ فرآیند یادگیری ۱۱
۲-۱-۳-۷ فرآیند تولید نمونه‌ی جدید ۱۲
۲-۱-۴ انواع معماری مدل‌های پخشی نویززدای احتمالاتی ۱۲
۲-۱-۴-۱ مدل پخشی نهان ۱۳
۲-۱-۴-۲ مدل پخشی مبتی بر مبدل ۱۵
۲-۲ رویکردهای افزایش سرعت ۱۶
۲-۲-۱ رویکردهای کاهش تعداد گام‌ها ۱۸
۲-۲-۲ رویکردهای کاهش محاسبات هر گام ۱۹
۲-۲-۲-۱ گسسته‌سازی ۲۰
۲-۲-۲-۲ کاهش توکن‌ها ۲۱
۲-۲-۲-۲-۱ روش ادغام توکن‌ها ۲۲
۲-۲-۲-۲-۲ روش همجوشی توکن‌ها ۲۴
۲-۲-۲-۳ استفاده از حافظه‌ی پنهان ۲۵
۲-۲-۲-۳-۱ روش استفاده از حافظه‌ی پنهان عمیق ۲۶
۲-۲-۲-۳-۲ مدل پخشی سریع‌تر ۲۸
۲-۲-۲-۳-۳ رویکرد گیت T ۳۰
۲-۳ جمع‌بندی ۳۲

۳ روش پیشنهادی ۳۳
۳-۱ روش پایه ۳۴
۳-۲ تطبیقی کردن روش ادغام توکن‌ها ۳۶

۳-۳ استفاده از حافظه‌ی نهان در ادغام توکن‌ها ۳۹

۴ آزمایش‌ها و نتایج ۴۲
۴-۱ مجموعه داده ۴۳
۴-۱-۱ مجموعه داده‌ی $ImageNet$ ۴۳
۴-۱-۲ مجموعه داده‌ی $COCO$ ۴۳
۴-۲ معیارهای ارزیابی ۴۵
۴-۳ تعیین ابرپارامترها ۴۸
۴-۴ تحلیل و بررسی نتایج ۵۲

۵ جمع‌بندی و راهکارهای آتی ۵۵

کتاب‌نامه ۵۸

واژه‌نامه‌ی فارسی به انگلیسی ۶۳

واژه‌نامه‌ی انگلیسی به فارسی ۶۶

فهرست تصاویر

شکل | صفحه

۲-۱ روند کلی یک مدل پخشی ۶
۲-۲ زنجیره‌ی فرآیند تولید تصویر در مپناها ۷
۲-۳ معماری $pixelCNN++$ ۱۳
۲-۴ معماری شبکه‌ی یو ۱۴
۲-۵ معماری مدل‌های پخشی پنهان ۱۵
۲-۶ معماری مدل‌های مبتنی بر مبدل ۱۶
۲-۷ مشکل سه‌گانه‌ی مدل‌های مولد ۱۸
۲-۸ رویکردهای انتخاب توکن‌های مبدا و مقصد در ادغام توکن‌ها ۲۳
۲-۹ روال کار ساخت گراف دوبخشی و ادغام توکن‌ها ۲۳
۲-۱۰ محل قرارگیری بلوک‌های ادغام و رفع ادغام در بلوک مبدل در مدل‌های پخشی . . . ۲۴
۲-۱۱ مشاهدات تجربی مبنی بر وجود افزونگی زمانی در ویژگی‌های سطح بالا در مدل‌های پخشی ۲۷
۲-۱۲ کلیت روش استفاده از حافظه‌ی پنهان عمیق ۲۷
۲-۱۳ مشاهدات تجربی مبتنی بر تغییرات ناچیز کدگذار شبکه‌ی یوی مدل پخشی ۲۹
۲-۱۴ کلیت مدل پخشی سریع‌تر ۳۰
۲-۱۵ نمودار تفاضل تغییرات بلوک توجه متقابل در گام‌های زمانی متوالی ۳۱
۲-۱۶ تقسیم فرآیند نویززدایی به دو بخش برنامه‌ریزی معنایی و افزایش کیفیت ۳۱

۳-۱ تصویر نمادین تفاوت حالت انطباق‌پذیر و حالت ثابت ادغام توکن‌ها ۳۷
۳-۲ نمودار فاصله‌ی جاکارد مجموعه‌ی جفت‌های ادغام در روش ادغام توکن‌ها ۴۰
۳-۳ شمای کلی روش پیشنهادی و استفاده از حافظه‌ی پنهان در فرآیند ادغام توکن‌ها . . ۴۱

۴-۱ نمونه تصاویری از مجموعه داده‌ی $ImageNet$ ۴۴
۴-۲ نمونه تصاویری از مجموعه داده‌ی $COCO$ ۴۵
۴-۳ نمونه تصاویر تولیدی برای تنظیمات مختلف جدول ۴-۲ و به ازای ۵ کلاس مختلف از $ImageNet$ ۵۱
۴-۴ نمونه تصاویر تولیدی از ۵ کلاس مجموعه داده‌ی $ImageNet$ در سه حالت مدل پایه‌ی $Stable\ Diffusion\ v1.5$، همراه با ادغام توکن و همراه با روش پیشنهادی . ۵۳
۴-۵ نمونه تصاویر تولیدی از ۵ کلاس مجموعه داده‌ی $ImageNet$ در سه حالت مدل پایه‌ی $Stable\ Diffusion\ v2$، همراه با ادغام توکن و همراه با روش پیشنهادی . . ۵۴

فهرست جداول

جدول صفحه

۳-۱ نتایج اعمال روش ادغام توکن‌ها بر بلوک‌های مبدل لایه‌های مختلف شبکه‌ی یو . . . ۳۵

۴-۱ معیارهای کیفیت و زمان برای آستانه‌های مختلف در روش ادغام تطبیقی در مدل $Stable\ Diffusion\ v1.5$ ۴۹

۴-۲ گام‌های محاسبه در تنظیمات مختلف در نظر گرفته شده در روش ادغام توکن تطبیقی مبتنی بر حافظه‌ی پنهان ۵۰

۴-۳ عملکرد کلی روش ادغام توکن مبتنی بر حافظه‌ی پنهان در تنظیمات گام‌های محاسبه‌ی متفاوت ۵۱

۴-۴ مقایسه‌ی مدل پایه‌ی $Stable\ Diffusion\ v1.5$ و مدل ادغام توکن‌ها با روش پیشنهادی در بهترین تنظیمات آن ۵۲

۴-۵ مقایسه‌ی مدل پایه‌ی $Stable\ Diffusion\ v2$ و مدل ادغام توکن‌ها با روش پیشنهادی در بهترین تنظیمات آن ۵۲

فهرست اختصارات

عنوان اختصاری	عنوان کامل
مپنا	مدل پخشی نویززدای احتمالاتی

فصل اول

مقدمه

در سال‌های اخیر، مدل‌های مولد[1] به‌عنوان یکی از ابزارهای قدرتمند در یادگیری ماشین و هوش مصنوعی، انقلابی در تولید داده‌های جدید ایجاد کرده‌اند. این مدل‌ها قابلیت تولید داده‌های متنوع و باکیفیت در حوزه‌های گوناگون مانند تصویر، متن، صوت و ویدئو را دارند و با الگوریتم‌های پیشرفته‌ای که به کار می‌گیرند، امکان بازسازی یا تولید داده‌هایی را فراهم می‌کنند که مورد نیاز افراد متخصص در طیف متنوعی از حوزه‌ها است. در میان این مدل‌ها، مدل‌های پخشی[2] به‌عنوان یکی از جدیدترین و مؤثرترین خانواده‌های مدل‌های مولد حوزه‌ی تصویر معرفی شده‌اند که قابلیت‌های بی‌نظیری را در تولید این داده‌ها نشان داده‌اند و به عنوان بهترین مدل[3] در حوزه‌ی تولید تصویر جایگاه خود را تثبیت کرده‌اند.

مدل‌های پخشی با فرآیندی مبتنی بر تخریب تدریجی داده‌های واقعی و سپس بازسازی معکوس آن‌ها عمل می‌کنند. این مدل‌ها انواع مختلفی دارند از جمله مدل‌های پخشی مبتنی بر امتیاز[4]، مدل‌های پخشی نویززدای احتمالاتی[5] و مدل‌های پخشی مبتنی بر معادله دیفرانسیل[6]. در مدل‌های پخشی نویززدای احتمالاتی، داده‌های واقعی به‌تدریج و در طول یک زنجیره‌ی مارکفی نویزی می‌شوند و در نهایت به یک توزیع ساده، معمولاً توزیع گاوسی، تبدیل می‌شوند. سپس در فرآیند رو به عقب[7] با شروع از یک نویز خالص، نویززدایی داده به‌صورت معکوس انجام می‌شود و به تولید نمونه‌های جدید منجر می‌گردد. این رویکرد ساده اما قدرتمند، مدل‌های پخشی را به یکی از برجسته‌ترین ابزارها برای تولید داده‌های باکیفیت تبدیل کرده است.

به طور کلی در هر مدل مولد حوزه‌ی تصویر سه نیازمندی مطرح است و اگر مدلی توانایی برآوردن این سه نیازمندی را داشته باشد می‌تواند در تمامی حوزه‌ها و تمامی کاربردها به راحتی مورد استفاده قرار گیرد[۳۴]. این سه نیازمندی عبارتند از کیفیت تصاویر تولیدی، تنوع تصاویر تولیدی و سرعت فرآیند تولید نمونه. مدل‌های پخشی در مقایسه با سایر مدل‌های مولد حوزه‌ی تصویر دارای کیفیت و تنوع نمونه‌های تولیدی هستند اما نمونه‌برداری در این مدل‌ها به نسبت سایر مدل‌ها بسیار آهسته است[۳۴]. فرآیند نمونه‌برداری در این مدل‌ها به گام‌های متعددی نیاز دارد و در هر گام باید محاسبات پیچیده‌ای انجام شود. این موضوع باعث می‌شود زمان اجرای مدل‌های پخشی برای تولید نمونه‌های جدید طولانی باشد و این امر استفاده از آن‌ها را در کاربردهای بلادرنگ و تعاملی محدود می‌کند.

تحقیقات بسیاری در سال‌های اخیر برای غلبه بر این محدودیت و افزایش سرعت مدل‌های پخشی انجام شده است. این رویکردها را می‌توان به اشکال مختلف دسته‌بندی کرد اما یک دسته‌بندی رایج

[1] Generative Models
[2] Diffusion Models
[3] State of the Art
[4] Score-based Diffusion Models
[5] Denoising Diffusion Probabilistic Models
[6] Stochastic Differential Equation based Models
[7] Reverse Process

که در این پژوهش نیز مدنظر است تقسیم رویکردها بر اساس محل اثر آنهاست. رویکردهای افزایش سرعت می‌توانند با کاهش تعداد گام‌های مدل پخشی سرعت آن را افزایش دهند یا می‌توانند با کاهش حجم محاسباتی که در هر گام صورت می‌گیرد، سرعت را افزایش دهند[۱۸]. رویکردهایی نظیر تقطیر دانش[۸][۲۵] و مدل‌های پخشی نویززدای ضمنی[۹][۲۹] در دسته‌ی کاهش تعداد گام‌ها قرار می‌گیرند. در مقابل، روش‌هایی مانند فشرده‌سازی شبکه‌ی نویززدا[۱۰] و استفاده از حافظه‌ی پنهان[۱۱] در دسته‌ی کاهش محاسبات هر گام قرار می‌گیرند. این روش‌ها اغلب با حفظ کیفیت خروجی، به دنبال کاهش زمان محاسبات و افزایش سرعت اجرای مدل بوده‌اند.

در این پژوهش با تمرکز بر روش‌های بدون آموزش کاهش محاسبات در هر گام، روش پیشنهادی تحت عنوان ادغام توکن‌های تطبیقی مبتنی بر حافظه‌ی پنهان ارائه شده است. این روش با تمرکز بر نقاط ضعف موجود در روش ادغام توکن‌ها[۱۲] و همچنین با در نظر گرفتن این مشاهده که میزان تغییر بلوک‌های شبکه‌ی یوی نویززدا[۱۳] در گام‌های متوالی ناچیز و هموار است [۱۴][۱۸]، تلاش می‌کند تا از طریق ترکیب روش استفاده از حافظه‌ی پنهان و روش ادغام توکن‌ها تعداد توکن‌های ورودی در هر بلوک مبدل[۱۴] را کاهش دهد. و این کاهش منجر به کاهش محاسبات انجام شده در هر گام از شبکه و در نتیجه کاهش زمان نمونه‌برداری در عین حفظ کیفیت می‌شود.

نتایج آزمایشات انجام شده بر روی مجموعه داده‌ی $ImageNet$[۴] نشان می‌دهد که روش پیشنهادی نسبت به روش پایه‌ی ادغام توکن‌ها و نیز نسبت به مدل پخشی $Stable\ Diffusion\ V1.5$ [۲۳]کاهش زمانی با ضریب ۱.۲۴ دارد. این ضریب کاهش در مدل پخشی $Stable\ Diffusion\ V2$ به ۱.۳۱ می‌رسد و همچنین نشان داده می‌شود که این کاهش حداکثر کاهش زمانی است که در روش ادغام توکن‌ها امکان دارد و این امر با ثابت نگه داشتن کیفیت تصاویر تولیدی انجام شده است.

در ادامه این پژوهش، ابتدا مفاهیم پایه و مبانی نظری مرتبط با مدل‌های پخشی و روش‌های بهبود سرعت آن‌ها مورد بررسی قرار خواهد گرفت. سپس جزئیات روش پیشنهادی ارائه شده و مراحل پیاده‌سازی آن شرح داده می‌شود. در بخش‌های پایانی نتایج آزمایشات انجام‌شده تحلیل شده و کارایی روش پیشنهادی با سایر روش‌های موجود مقایسه می‌گردد. نهایتاً، راهکارهایی برای تحقیقات آینده در این زمینه ارائه خواهد شد.

[8] Knowledge Distillation
[9] Denoising Diffusion Implicit Models(DDIM)
[10] Network Compression
[11] Caching
[12] Token Merging
[13] Denoising U-net
[14] Transformer

فصل دوم

مدل‌های پخشی و رویکردهای افزایش سرعت

۲-۱ تعاریف

۲-۱-۱ مدل‌های مولد

مدل‌های مولد عمیق دسته‌ای از مدل‌های پرکاربرد و جدید در حوزه‌ی هوش مصنوعی هستند که تلاش می‌کنند تا توزیع اصلی داده‌ی نمونه‌برداری شده را مدل کنند و با استفاده از این توزیع انواع مختلفی از سوالات در خصوص داده و دسته‌های آن را پاسخ دهند. در واقع اصلی‌ترین هدف یک مدل مولد، استفاده از مشاهدات شبیه‌سازی شده یا نمونه برداری شده از یک توزیع برای تخمین آن توزیع و سپس تولید نمونه‌های جدید از توزیع تخمین زده شده می‌باشد. مدل‌های مولد کاربردهای متنوعی در یادگیری ماشین دارند، از جمله‌ی این کاربردها می‌توان به تولید عکس‌های باکیفیت، تولید قطعه‌های موسیقی و یا صوت، افزایش کارایی یادگیری نیمه نظارتی[۱]، تشخیص ناهنجاری[۲]، یادگیری تقلیدی و کمک به یادگیری تقویتی اشاره کرد. پیشرفت‌های اخیر این حوزه دو رویکرد کلی را دنبال می‌کنند: رویکردهای مبتنی بر درست‌نمایی[۳] و شبکه‌های مولد تقابلی[۴] [۳۰]. در رویکردهای مبتنی بر درست‌نمایی از لگاریتم درست‌نمایی یا خطاهای جایگزین به عنوان تابع هدف یادگیری استفاده می‌شود؛ در حالیکه در شبکه‌های مولد تقابلی تلاش می‌شود تا یادگیری با یک رویکرد تقابلی بین دو شبکه‌ی تمایزگر[۵] و تولیدکننده[۶] صورت گیرد.

با وجود عملکرد قابل توجه مدل‌های مبتنی بر درست‌نمایی وشبکه‌های مولد تقابلی، این مدل‌ها به واسطه‌ی نوع تعریفشان درگیر برخی محدودیت‌های ذاتی هستند. به عنوان مثال مدل‌های مبتنی بر درست‌نمایی برای ساخت توزیع‌های نرمال شده نیازمند به کارگیری معماری‌های خاص یا استفاده از توابع خطای جایگزین (مانند حد پایین مشاهده[۷] در خودکدگذارهای تغییراتی[۸]) در فرآیند یادگیری هستند. شبکه‌های مولد تقابلی مشکلات موجود در مدل‌های مبتنی بر درست‌نمایی را ندارند اما فرآیند یادگیری آنها به خاطر ماهیت تقابلی آن می‌تواند دچار ناپایداری شود. همچنین تابع هدف مدل‌های مولد تقابلی برای مقایسه و ارزیابی انواع این مدل‌ها چندان مناسب نیست[۳۰].

۲-۱-۲ مدل‌های پخشی

مدل‌های پخشی[۹] یک دسته‌ی نوظهور از مدل‌های مولد هستند که با توجه به عملکرد مناسب آنها در حوزه‌های تصویر، صوت، متن و گراف به سرعت مورد استقبال قرار گرفتند. وجه مشترک همه‌ی مدل‌های

[1] Semi-supervised learning
[2] Anomaly
[3] Likelihood
[4] Generative adversarial networks
[5] Discriminatior
[6] Generator
[7] Evidence lower bound
[8] Variational autoencoders
[9] Diffusion models

پخشی این است که در یک فرآیند رو به جلو[10] داده را با نویز تخریب می‌کنند ودر نتیجه از توزیع داده به یک توزیع دلخواه قابل محاسبه می‌رسند و سپس در یک مسیر رو به عقب که توسط مدل یاد گرفته می‌شود، تلاش می‌کنند تا داده را بازسازی و از یک نمونه از توزیع دلخواه به یک نمونه از توزیع داده‌ها برسند. این موضوع درشکل ۲-۱ نشان داده شده است.

شکل ۲-۱: روند کلی یک مدل پخشی

در هر مدل پخشی یک فرآیند ثابت گام به گام رو به جلو برای تخریب داده و یک فرآیند معکوس برای تولید یک نمونه داده از نویز وجود دارد[۱۱].

با وجود اینکه چهارچوب گفته شده در بند قبل در همه‌ی مدل‌های پخشی وجود دارد؛ رویکردهای مختلفی برای فرموله‌سازی و پیاده‌سازی این روش وجود دارد. در مدل‌های مولد مبتنی بر تابع امتیاز[11] تلاش می‌شود با استفاده از گرادیان لگاریتم توزیع داده، که به آن تابع امتیاز[12] می‌گوییم، و یک فرآیند گام به گام حرکت در جهت این گرادیان نمونه‌ی جدید تولید کنیم[۳۱]. از طرفی در مدل‌های پخشی نویززدای احتمالاتی[۹] فرآیند تخریب داده و تولید نمونه با یک زنجیره‌ی مارکفی احتمالاتی مدل می‌شود و در نهایت در مدل‌های پخشی مبتنی بر معادلات دیفرانسیل تصادفی[۳۲]، فرآیند تخریب به صورت یک معادله دیفرانسیل تصادفی پیوسته مدل می‌شود که با پیدا کردن معکوس این معادله و گسسته‌سازی آن امکان تولید نمونه‌ی جدید از توزیع داده‌ها فراهم می‌گردد.

می‌توان نشان داد که مدل‌های پخشی بسیاری از محدودیت‌های مدل‌های مولد دیگر مانند نیاز به تمایزگر در شبکه‌های مولد تقابلی، نیاز به مقیدسازی شبکه در جریان‌های نرمال‌ساز[13] و نیاز به تنظیم توزیع‌های پسین در خودکدگذارهای تغییراتی را ندارند و قابلیت تولید تصاویر باکیفیت و متنوع را دارند. مشکل اصلی در مدل‌های پخشی فرآیند نمونه‌برداری طولانی آنها است که باعث عدم کارایی این مدل‌ها در کاربردهای تعاملی می‌شود[۳۴]. با بهبود این چالش می‌توان اطمینان حاصل کرد که این دسته از مدل‌های مولد از توانایی همه‌جانبه‌ای برای به کارگیری در حوزه‌های مختلف برخوردار هستند.

[10] Forward process

[11] Score-based generative models

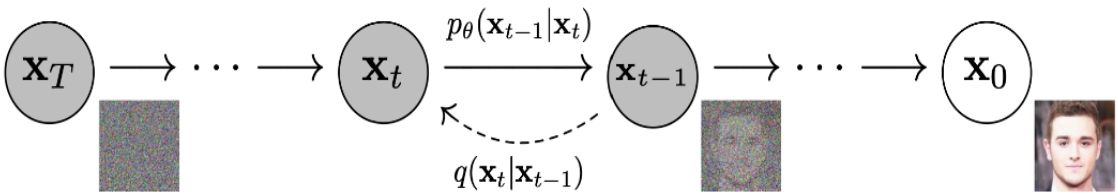
[12] Score function

[13] Normalizing flows

۲-۱-۳ مدل‌های پخشی نویززدای احتمالاتی

۲-۱-۳-۱ تعریف مدل

مدل‌های پخشی نویززدای احتمالاتی[۱۴][۹] یا به اختصار مپناها مدل‌هایی مبتنی بر متغیر پنهان[۱۵] هستند که از یک زنجیره‌ی مارکفی [۱۶]برای فرآیند پخشی استفاده می‌کنند. رویکرد کلی این مدل‌ها به این صورت است که در یک مسیر رو به جلو[۱۷] هر توزیع پیچیده‌ای از داده‌ها را می‌گیرد و آن را به یک توزیع اولیه که ساده و قابل محاسبه است تبدیل می‌کند و سپس یک مسیر رو به عقب[۱۸] محدود به زمان را یاد می‌گیرد که نمونه هایی از توزیع اولیه را به نمونه‌های توزیع داده تبدیل کند و این همان ایده‌ی کلی مدل مولد پخشی می‌باشد. تصویر ۲-۲ این فرآیند را نشان می‌دهد. بنابراین این مدل سه مشخصه‌ی اصلی دارد: مسیر رو به جلو، مسیر رو به عقب و تابع مدل مولد که در ادامه هر یک را توضیح می‌دهیم.



شکل ۲-۲: زنجیره‌ی فرآیند تولید تصویر در مپناها

مدل گرافی جهت‌دار یک مپنا که به صورت یک زنجیره‌ی مارکف روبه جلو و معکوس آن می‌باشد[۹].

۲-۱-۳-۲ مسیر رو به جلو

با داشتن یک نمونه‌ی اولیه‌ی $x_0 \backsim q(x)$ فرآیند رو به جلو فرآیندی است که در آن در T گام مقداری نویز گاوسی به داده اضافه می‌کنیم و دنباله‌ی $x_0, x_1, ..., x_T$ شامل نمونه‌های نویزی را تولید می‌کنیم. اندازه‌ی این گام‌ها با هایپرپارامتری به نام برنامه‌ی واریانس[۱۹] و نماد β_t کنترل می‌شود و برای این پارامتر داریم: $\{\beta_t \in (0,1)\}_{t=1}^{T}$. بر این اساس رابطه‌ی مسیر رو به جلو به صورت زیر می‌باشد[۹]:

[14] Denoising diffusion probabilistic models(DDPMs)
[15] Latent variable models
[16] Markov Chain
[17] Forward Path
[18] Backward Path
[19] Variance Schedule

$$q(\mathbf{x}_t|\mathbf{x}_{t-1}) = \mathcal{N}(\mathbf{x}_t; \sqrt{1-\beta_t}\mathbf{x}_{t-1}, \beta_t\mathbf{I}) \quad q(\mathbf{x}_{1:T}|\mathbf{x}_0) = \prod_{t=1}^{T} q(\mathbf{x}_t|\mathbf{x}_{t-1}) \tag{۱-۲}$$

که $\mathcal{N}$ نماد توزیع نرمال یا گاوسی است و $\mathbf{I}$ نیز ماتریس همانی است. نمونه داده‌ی x_0 به تدریج و با بزرگتر شدن t تخریب می‌شود تا در نهایت و با $T \to \infty$ به یک نمونه از توزیع گاوسی همسانگرد[۲۰] تبدیل می‌شود.

یک ویژگی حائز اهمیت مسیر رو به جلو این است که می‌توان به جای استفاده از فرم زنجیره‌ی مارکفی آن از یک فرم بسته استفاده کرد به این معنا که برای یافتن x_t در زمان دلخواه t می‌توان از رابطه‌ی زیر کمک گرفت:

$$q(x_t|x_0) = \mathcal{N}(x_t; \sqrt{\overline{\alpha}_t}x_0, (1-\overline{\alpha}_t)I) \tag{۲-۲}$$

که $\overline{\alpha}_t := \prod_{s=1}^{t} \alpha_s$ و $\alpha_t := 1 - \beta_t$ تعریف می‌شود و β_t همان برنامه‌ی واریانس است.

۲-۱-۳-۳ مسیر رو به عقب

برای یادگیری توزیع مولد نیاز است که بتوانیم برعکس مسیر رو به جلو را یاد بگیریم و این به این معناست که باید بتوانیم از یک توزیع دلخواه قابل محاسبه به توزیع داده برسیم. بنابراین با دانستن توزیع اولیه $x_T \backsim \mathcal{N}(0, \mathbf{I})$ داریم:

$$\begin{aligned} p_\theta(\mathbf{x}_{0:T}) &= p(\mathbf{x}_T) \prod_{t=1}^{T} p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) \\ p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) &= \mathcal{N}(\mathbf{x}_{t-1}; \mu_\theta(\mathbf{x}_t, t), \mathbf{\Sigma}_\theta(\mathbf{x}_t, t)) \end{aligned} \tag{۳-۲}$$

که در این رابطه $p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$ تخمین حاصل از یک شبکه‌ی عصبی برای هسته‌ی گذر رو به عقب و $\mathcal{N}$ نماد توزیع نرمال و $\mu_\theta(\mathbf{x}_t, t)$ میانگین تخمین زده‌شده‌ی این توزیع و $\mathbf{\Sigma}_\theta(\mathbf{x}_t, t)$ نیز ماتریس کوواریانس آن می‌باشد. می‌توان نشان داد که اگر فرآیند پخش گاوسی باشد و برنامه‌ی واریانس فرآیند رو به جلوی β خیلی کوچک باشد آنگاه معکوس فرآیند پخش نیز گاوسی می‌باشد[۵]. و بنابراین هر چقدر که طول این زنجیره بیشتر باشد β کوچکتر می‌شود و می‌توان از تابع گاوسی برای مسیر معکوس نیز کمک گرفت.

[20] Isotropic

در فرآیند یادگیری با توجه به اینکه هسته‌ی گذر[۲۱] زنجیره‌ی مارکفی یک تابع گاوسی است، برای ساخت زنجیره تنها نیاز به میانگین و کوواریانس این تابع گاوسی داریم و در ادامه سعی می‌کنیم آنها را تخمین بزنیم.

۲-۱-۳-۴ تعریف تابع خطا

تابع مدل مولد تولید شده توسط مپنا به صورت زیر می‌باشد:

$$p_\theta(x_0) = \int d\mathbf{x}_{1:T} p_\theta(\mathbf{x}_{0:T}) \tag{۲-۴}$$

با توجه به این تابع احتمال، آموزش شبکه معادل بیشینه کردن لگاریتم درست‌نمایی[۲۲] تابع مدل مولد و یا کمینه کردن منفی این تابع است. در اینجا برای آموزش به جای بهینه سازی مستقیم لگاریتم درست‌نمایی از بهینه سازی حد تغییراتی[۲۳] آن کمک می‌گیریم که به صورت زیر قابل تعریف است:

$$\begin{aligned}
\mathbb{E}[-\log p_\theta(\mathbf{x}_0)] &\leq \mathbb{E}_q\left[-\log \frac{p_\theta(\mathbf{x}_{0:T})}{q(\mathbf{x}_{1:T}|\mathbf{x}_0)}\right] \\
&= \mathbb{E}_q\left[-\log p(\mathbf{x}_T) - \sum_{t\geq 1} \log \frac{p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)}{q(\mathbf{x}_t|\mathbf{x}_{t-1})}\right] \\
&=: L
\end{aligned} \tag{۲-۵}$$

که در این رابطه $\mathbb{E}_q$ نماد محاسبه‌ی امیدریاضی بر روی توزیع رو به جلوی q و رابطه‌ی سطر دوم نیز با استفاده از رابطه‌ی ۲-۳ قابل به دست آوردن از سطر اول می‌باشد.

۲-۱-۳-۵ تغییر پارامتر تابع خطا

از آنجایی که ترکیب مسیر رو به جلو و مسیر معکوس شبیه به یک کدگذار تغییراتی است، می‌توان تابع هزینه را به صورت زیر بازنویسی کرد:

$$L = L_0 + \sum_{t>1} L_{t-1} + L_T \tag{۲-۶}$$

[21] Transition kernel
[22] Log-likelihood
[23] Variational bound

و در تعریف هر کدام از سه عبارت L_0، L_{t-1} و L_T داریم:

$$L_0 = -\log p_\theta(x_0|x_1) \tag{۷-۲}$$

$$L_{t-1} = \mathcal{D}_{KL}(q(x_{t-1}|x_t, x_0)\|p_\theta(x_{t-1}|x_t)) \tag{۸-۲}$$

$$L_T = \mathcal{D}_{KL}(q(x_T|x_0)\|p(x_T)) \tag{۹-۲}$$

با توجه به ثابت بودن عبارت L_T نسبت به θ این عبارت در بهینه‌سازی در نظر گرفته نمی‌شود و همچنین با توجه به غالب بودن عبارت $\sum_{t>1} L_{t-1}$ نسبت به L_0 از این عبارت نیز صرف نظر می‌شود[۹]. از آنجایی که در صورت کوچک بودن β_t هر دوی $p_\theta(x_{t-1}|x_t))$ و $q(x_{t-1}|x_t, x_0)$ توزیع هایی گاوسی هستند در نتیجه فاصله‌ی KL[۲۴] آنها را می‌توان به فرم تفاضل میانگین آنها در نظر گرفت و در نتیجه برای معادله ی (۲-۸) داریم:

$$L_{t-1} = \mathbb{E}_q\left[\frac{1}{2\sigma_t^2}\|\tilde{\mu}_t(x_t, x_0) - \mu_\theta(x_t, t)\|^2\right] + C \tag{۱۰-۲}$$

که در این رابطه σ_t^2 نمایانگر واریانس توزیع رو به عقب در گام t و $\tilde{\mu}_t$ میانگین پسین[۲۵] فرآیند رو به جلو و μ_θ میانگین تخمین زده‌شده‌ی توزیع رو به عقب و C نیز ثابتی مستقل از θ است. این رابطه در واقع نشان می‌دهد که در فرآیند آموزش یک مدل پخشی بهینه سازی خطا شامل تخمین میانگین پسین فرآیند رو به جلو می‌باشد. لازم است اشاره کنیم که در مقاله‌ی اصلی[۹] مقدار واریانس ثابت در نظر گرفته شده است. از طرفی با تغییر پارامتر $x_t(x_0, \epsilon) = \sqrt{\bar{\alpha}_t}x_0 + \sqrt{(1-\bar{\alpha}_t)}\epsilon$ برای $\epsilon \sim \mathcal{N}(0, I)$ و قراردهی آن در فرمول معادله‌ی (۲-۱۰) می‌توان نشان داد که تخمین $\mu_\theta(x_t, t)$ به صورت زیر قابل پارامتری سازی است:

$$\mu_\theta(x_t, t) = \tilde{\mu}_t\left(x_t, \frac{1}{\sqrt{\bar{\alpha}_t}}\left(x_t - \sqrt{1-\bar{\alpha}_t}\epsilon_\theta(x_t)\right)\right) = \frac{1}{\sqrt{\alpha_t}}(x_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_\theta(x_t, t) \tag{۱۱-۲}$$

[24] KL-divergence

[25] Posterior mean

که ϵ_θ یک تخمین از نویز موجود در x_t می‌باشد و در نتیجه برای محاسبه ی یک نمونه ی x_{t-1} تنها کافیست که عبارت زیر را محاسبه کنیم:

$$x_{t-1} = \frac{1}{\sqrt{\alpha_t}}(x_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_\theta(x_t, t) + \sigma_t z \tag{۱۲-۲}$$

که z از توزیع گاوسی با میانگین صفر و کوواریانس همانی است.

این تغییر پارامتر نشان می‌دهد که برای آموزش یک فرآیند رو به عقب می‌توان از تخمین $\tilde{\mu}_t$ با تابع میانگین وابسته به زمان μ_θ استفاده کرد و یا مستقیما نویز اضافه شده به داده را با تخمین ϵ_θ محاسبه کرد.

۲-۱-۳-۶ فرآیند یادگیری

با توجه به تغییر پارامتر انجام شده در قسمت قبل و همچنین به کارگیری آن درمعادله‌ی (۲-۶) می‌توان به تابع هزینه ای رسید که کاملا نسبت به θ مشتق پذیر است و برای آموزش مناسب است اما برای سادگی پیاده‌سازی و افزایش کیفیت نمونه‌های تولیدی می‌توان تابع هزینه را به شکل زیر نیز بازنویسی کرد:

$$L_{simple}(\theta) = \mathbb{E}_{t,x_0,\epsilon}\left[\left\|\epsilon - \epsilon_\theta(\sqrt{\bar{\alpha}_t}x_0 + \sqrt{1-\bar{\alpha}_t}\epsilon, t)\right\|^2\right] \tag{۱۳-۲}$$

که t از یک توزیع یونیفرم بین ۱و T می‌آید و واریانس β_t در این فرآیند یک پارامتر ثابت بدون یادگیری می‌باشد و همچنین $\alpha_t := 1 - \beta_t$ و $\overline{\alpha}_t := \prod_{s=1}^{t} \alpha_s$ تعریف می‌شود[۹]. الگوریتم ۱ فرآیند آموزش یک مپنا را نشان می‌دهد.

الگوریتم ۱ آموزش[۹]

۱: مراحل زیر را تکرار کن

۲: $\mathbf{x}_0 \sim q(\mathbf{x}_0)$

۳: $t \sim Uniform(\{1, \ldots, T\})$

۴: $\boldsymbol{\epsilon} \sim \mathcal{N}(0, \mathbf{I})$

۵: گام نزول در جهت گرادیان را روی تابع زیر اعمال کن

$$\nabla_\theta \|\boldsymbol{\epsilon} - \epsilon_\theta(\sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}, t)\|^2$$

۶: تا زمانی که همگرا شود

۲-۱-۳-۷ فرآیند تولید نمونه‌ی جدید

همانطور که در الگوریتم ۲ نشان داده شده است؛ برای نمونه برداری ابتدا یک نمونه از توزیع $\mathcal{N}(0, \mathbf{I})$ تولید می‌کنیم سپس به صورت معکوس هر بار نمونه را به شبکه می‌دهیم و نمونه‌ی نویززدایی شده را دریافت می‌کنیم. این فرآیند تا جایی ادامه پیدا می‌کند که به یک نمونه از توزیع $q(\mathbf{x}_0)$ برسیم.

الگوریتم ۲ نمونه‌برداری[۹]

۱: $\mathbf{x}_T \sim \mathcal{N}(0, \mathbf{I})$

۲: برای $t = T, \ldots, 1$ مراحل زیر را انجام بده:

۳: $\mathbf{z} \sim \mathcal{N}(0, \mathbf{I})$ اگر $t > 1$، در غیر این صورت $\mathbf{z} = 0$

۴: $\mathbf{x}_{t-1} = \frac{1}{\sqrt{\alpha_t}}\left(\mathbf{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_\theta(\mathbf{x}_t, t)\right) + \sigma_t \mathbf{z}$

۵: پایان حلقه

۶: مقدار $\mathbf{x}_0$ را برگردان.

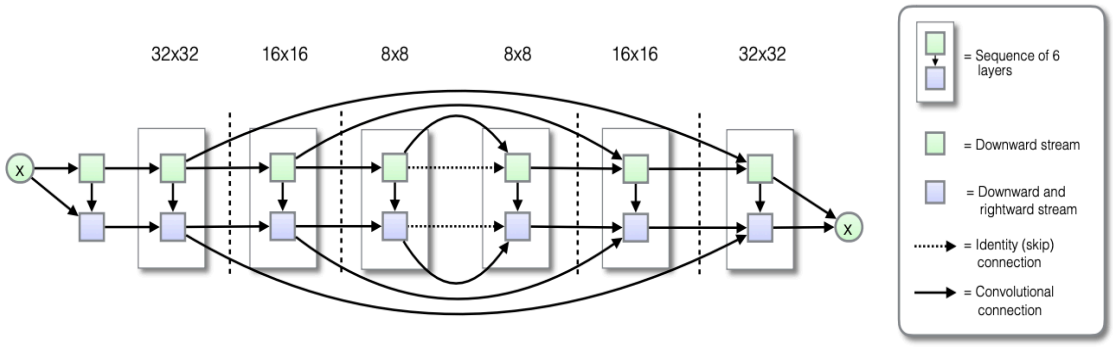
در بخش بعدی سراغ انواع معماری‌ها و پیاده‌سازی‌های مپناها می‌رویم.

۲-۱-۴ انواع معماری مدل‌های پخشی نویززدای احتمالاتی

در ساختار مپناها از هسته ی اصلی $pixelCNN++$ [۲۶] استفاده شده است که در واقع نوعی شبکه‌ی یو[۲۶] با اتصالات گسترده‌ی باقی‌مانده‌ای می‌باشد. شکل ۲-۳ ساختار کلی شبکه‌ی به کار گرفته شده را

[26] U-net

نشان می‌دهد.



شکل ۲-۳: معماری $pixelCNN++$

شبکه‌ی $pixelCNN++$ که نوعی شبکه‌ی یو با اتصالات باقی‌مانده‌ای گسترده است به عنوان شبکه‌ی اصلی در مپناها استفاده می‌شود[۲۶].

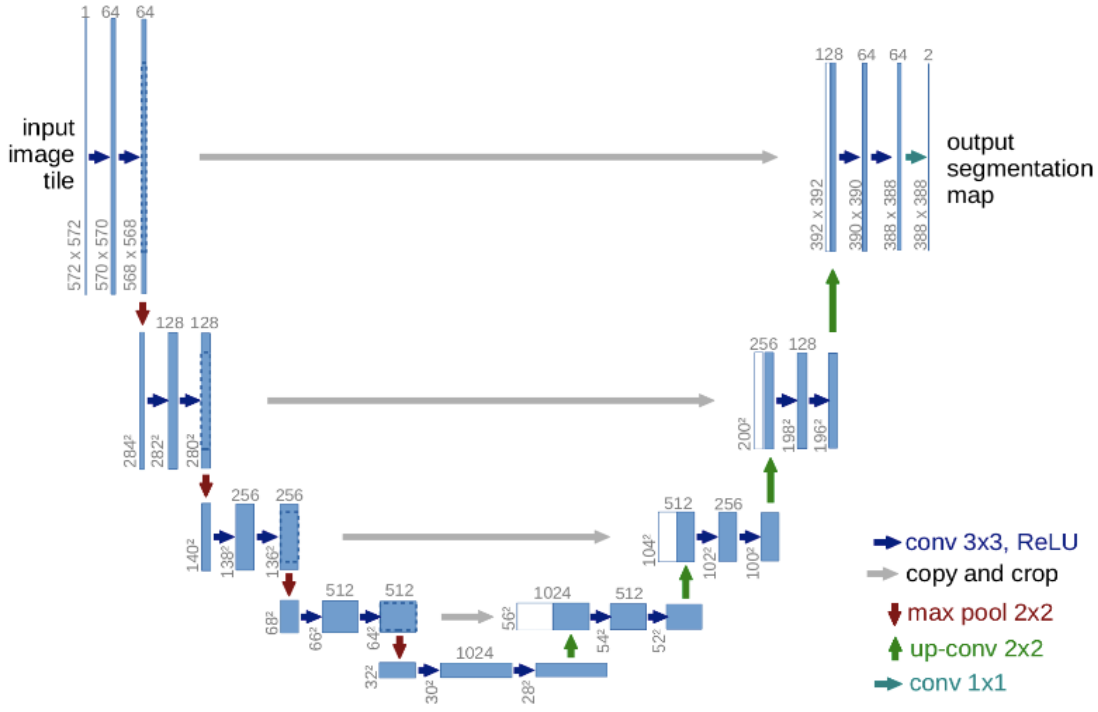
تا قبل از ارائه‌ی مدل پخشی نهان[۲۷][۲۳]عمده‌ی ساختار ارائه شده برای مپنا به تنهایی بر پایه‌ی شبکه‌ی یو[۲۴] یا یکی از انواع این شبکه بود. شبکه‌ی یو یک ساختار کدگذار-کدگشا[۲۸] است که ابتدا برای مساله‌ی قطعه‌بندی تصاویر پزشکی ارائه شد. ویژگی خاص این شبکه نسبت به سایر ساختارهای کدگذار-کدگشا وجود اتصالات پرشی[۲۹] می‌باشد. این اتصالات با هدف ترکیب اطلاعات سطح بالا و اطلاعات محلی و نیز افزایش رزولوشن تصویر خروجی استفاده شدند. شکل ۲-۴ ساختار یک شبکه‌ی یو را نشان می‌دهد.

۲-۱-۴-۱ مدل پخشی نهان

استفاده از شبکه‌ی یو به تنهایی، چالش‌های زیاد دارد؛ از جمله‌ی این مشکلات می‌توان به هزینه‌ی محاسباتی و سنگین بودن شبکه‌ی یو اشاره کرد که این موضوع با افزایش رزولوشن تصویر می‌تواند هزینه‌ی زیادی را ایجاد کند. از طرفی در اکثر مپناها فرآیند آموزش و نمونه‌برداری شبکه‌ی یو در فضای پیکسلی انجام می‌شود که مارا نیازمند تعریف یک شبکه‌ی متناظر به ازای هر رزولوشن و یا تغییر ابعاد تصویر می‌کند. علاوه بر موارد ذکر شده در شبکه‌های یوی به کار گرفته شده در فرآیند مپنا امکان شرطی‌سازی انعطاف پذیر وجود ندارد. این شبکه امکان انجام شرطی‌سازی بر اساس نام کلاس‌ها یا نویز ابتدایی تصویر را دارد ولی امکان دریافت متن و یا نقشه‌ی معنایی[۳۰] و پردازش آنها به عنوان شرط را بدون بهینه‌سازی در معماری ندارد[۲۳].

مدل‌های پخشی نهان با تمرکز بر رفع نقاط ضعف اشاره شده در مدل‌های شبکه‌ی یوی ابتدایی ارائه

[27] Latent Diffusion model
[28] Encoder-Decoder
[29] Skip Connection
[30] Semantic Map



شکل ۲-۴: معماری شبکه‌ی یو

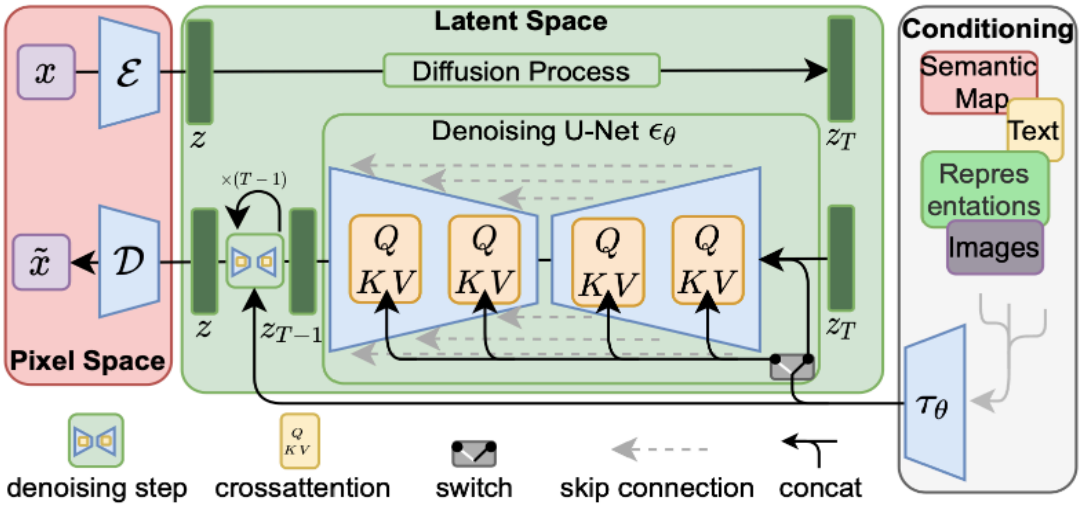
معماری کلی شبکه‌ی یو که اولین بار برای مساله‌ی قطعه‌بندی معنایی تصاویر پزشکی ارائه شد و شامل سه بخش کدگذار، کدگشا و اتصالات پرشی است[۲۴].

شدند. این مدل‌ها تلاش می‌کنند که اولا با انتقال تصویر از فضای پیکسلی به فضای پنهان با ابعاد کمتر از حجم محاسبات بکاهند و ثانیا با اضافه کردن بلوک‌های توجه متقابل[۳۱] انعطاف‌پذیری مدل را به واسطه‌ی امکان شرطی‌سازی بر روی هر شکلی از شرط شامل متن یا نقشه‌ی معنایی افزایش دهند. خروجی این تغییرات باعث تولید تصاویر با کیفیت بهتر و در رزولوشن‌های بالاتر است.

شکل ۲-۵ معماری این مدل را نشان می‌دهد. این مدل با به‌کارگیری یک خودکدگذار تغییراتی از پیش آموزش داده شده، ابتدا تصویر را از فضای پیکسلی به فضای پنهان می‌برد. در فضای پنهان، فرآیند یادگیری و نمونه‌برداری مشابه هر مدل مپنای دیگری انجام می‌شود و معماری اصلی شبکه‌ی نویززدا نیز یک شبکه‌ی یو با اتصالات پرشی و بلوک‌های توجه متقابل است. این مدل با استفاده از بلوک‌های توجه متقابل شرط را روی فرآیند نویززدایی اعمال می‌کند و در نهایت این مدل پس از پایان مراحل نویززدایی تصویر را از فضای پنهان به فضای پیکسلی باز می‌گرداند.

مدل‌های $Stable\ Diffusion$ که در این پروژه به کار گرفته شده اند؛ از این معماری تبعیت می‌کنند. که در فصل‌های پیش رو در خصوص جزئیات پیاده‌سازی آنها بیشتر صحبت خواهیم کرد.

[31] Cross-Attention Blocks



شکل ۲-۵: معماری مدل‌های پخشی پنهان

در این شکل شبکه‌ی ϵ کدگذار خود کدگذار تغییراتی است که تصویر ورودی x را دریافت می‌کند و آن را به تصویر در فضای پنهان z تبدیل می‌کند. همچنین τ_θ یک کدگذار مختص دامنه‌ی مساله است که برای تبدیل شرط به یک بازنمایی قابل درک توسط مدل به کار گرفته می‌شود. همچنین D نیز کدگشای خودکدگذار تغییراتی است که تصویر نویززدایی شده در فضای پنهان را به تصویر در فضای پیکسلی می‌برد[۲۳].

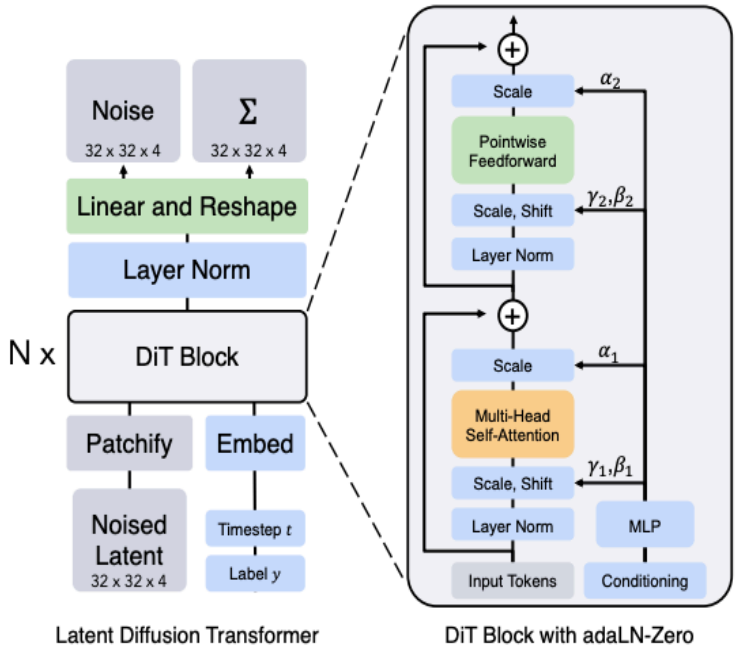
۲-۱-۴-۲ مدل پخشی مبتنی بر مبدل

در مدل‌های پخشی مبتنی بر مبدل[۳۲] [۲۱]به جای استفاده از شبکه‌ی یو به عنوان شبکه‌ی نویززدا از ساختار مبدل تصویری[۳۳] استفاده می‌شود. انتخاب این مبدل‌ها به عنوان هسته‌ی اصلی مدل‌های پخشی از این جهت است که هم می‌توانند وابستگی‌های با دامنه‌ی طولانی[۳۴] قسمت‌های مختلف تصویر را مدل کنند و هم اینکه نسبت به شبکه‌ی یو ساختارهای مقیاس‌پذیرتری هستند. همچنین به کارگیری این مبدل‌ها در انواعی از دامنه‌ها مانند پردازش زبان طبیعی و سایر وظایف حوزه‌ی بینایی کارایی آنها را نسبت به سایر معماری‌ها نشان می‌دهد و در نهایت وجود مکانیزم توجه به خود در مدل‌های مبدل باعث می‌شود که وابستگی‌های موجود در نواحی دور یا نزدیک تصویر به خوبی مدل شوند.

این مدل، مشابه مدل پخشی پنهان در فضای پنهان حاصل از یک خودکدگذار تغییراتی نویززدایی می‌کند. این مدل به جای استفاده از یک شبکه‌ی نویززدایی مبتنی بر شبکه‌ی یو از یک ساختار مطابق شکل ۲-۶ استفاده می‌کند. در این حالت ابتدا تصویر ورودی به فضای پنهان برده می‌شود و در آنجا مدل تصویر پنهان را به تعدادی قطعه[۳۵] تقسیم می‌کند که توسط مبدل پردازش می‌شوند. همچنین در این

[32] Diffusion Transformer Models
[33] Vision Transformer
[34] Long-range Dependency
[35] Patch

مدل از یک رویکرد به نام $adaLN - zero$ برای اضافه کردن نام کلاس و شرطی‌سازی استفاده می‌شود. این رویکرد برخلاف مدل‌های پخشی پنهان که از یک بلوک توجه متقابل استفاده می‌کند، ابتدا شرط یا بازنمایی آن را از یک شبکه‌ی تماما متصل عبور می‌دهند و سپس آن را در قالب پارامترهای تغییر مقیاس[۳۶] به لایه‌های نرمالسازی لایه‌ای اضافه می‌کند.



شکل ۲-۶: معماری مدل‌های مبتنی بر مبدل

همانطور که شکل نشان می‌دهد، ورودی‌های این مدل تصویر و زمان و برچسب کلاس هستند. تصویر ابتدا قطعه قطعه می‌شود و سپس وارد بلوک مبدل شده و در نهایت نیز یک تبدیل خطی روی آن انجام می‌شود تا با ترکیب قطعات مختلف تصویر نهایی ایجاد شود[۲۱].

این مدل بر اساس اندازه‌ی قطعه، تعداد سرهای[۳۷] مکانیزم توجه، تعداد لایه‌ها و اندازه‌ی لایه‌ی پنهان می‌تواند انواع گوناگونی داشته باشد. با توجه به آزمایشات انجام شده در مقاله [۲۱] مدل $DIT - XL$ با ۲۸ لایه، اندازه قطعه‌ی ۲، اندازه‌ی لایه‌ی پنهان ۱۱۵۲ و تعداد سرهای ۱۶ بهترین عملکرد را از خود ارائه می‌دهد.

۲-۲ رویکردهای افزایش سرعت

با وجود توسعه‌ی حجم زیادی از مدل‌های مولد در سال‌های اخیر برای انواعی از کاربردها از جمله تصویر، صدا، متن و گراف ها، همه‌ی مدل های موجود فعلی قابلیت ارضای سه نیاز اصلی یک مدل مولد به کار

[36] Scale Shift

[37] Head

گرفته شده در مسائل واقعی را ندارند[۳۴]. این سه نیاز اصلی عبارتند از:

۱. تولید نمونه‌های با کیفیت

۲. پوشش مد و تنوع نمونه‌های تولیدی

۳. نمونه برداری سریع و به صرفه از نظر محاسباتی

به عنوان مثال اکثر مدل‌های تولید تصویر فعلی روی تولید نمونه‌های با کیفیت تمرکز دارند اما تنوع داده‌ها نیز موضوع مهمی در کاهش تاثیرات اجتماعی منفی مدل‌های مولد است. همچنین برای کاربردهای تعاملی نیاز به نمونه برداری سریع یک مساله‌ی حائز اهمیت است. این چالش در ادبیات مدل‌ها مولد به عنوان مشکل سه‌گانه‌ی یادگیری مولد[۳۸] شناخته می‌شود[۳۴]. هر کدام از مدل‌های مولد در حل بخشی از مشکلات یادگیری مولد تبحر دارند و بخش‌هایی از این سه‌گانه به عنوان ضعف آنها محسوب می‌شود. به عنوان مثال شبکه‌های مولد تقابلی قابلیت تولید نمونه‌های با کیفیت در زمان سریع را دارند اما پوشش مد ضعیفی دارند. از طرفی خودکدگذار های تغییراتی به خوبی مدهای داده را پوشش می‌دهند و سرعت نمونه‌برداری مناسبی دارند اما کیفیت نمونه‌های تولیدی آنها کم است. برای مقایسه‌ی عملکرد مدل‌های پخشی با سایر مدل‌های مولد خوب است بدانیم که این مدل‌ها قابلیت رفع این مشکل سه گانه را دارند یاخیر. مدل‌های پخشی با ظهور خود توانستند از شبکه‌های مولد تقابلی در تولید نمونه‌های با کیفیت پیشی بگیرند و همچنین با تغییر و تنظیم این مدل‌ها و دستیابی به درست‌نمایی بهینه، قابلیت پوشش مدهای مختلف داده را دارند اما نمونه‌برداری از این مدل‌ها نیازمند صدها بار استفاده از شبکه می‌باشد و همین امر سرعت نمونه‌برداری را کاهش می‌دهد و به‌کارگیری آنها در کاربردهای مختلف را هزینه‌بر و دشوار می‌سازد. شکل ۲-۷ این مشکل سه گانه و حوزه ی عملکرد مناسب هر مدل را نشان می‌دهد.

با توجه به دسته‌بندی ذکر شده در بند قبل، اصلی‌ترین نقطه ضعف عمده‌ی مدل‌های پخشی سرعت نمونه‌برداری کم آنها است و در همین راستا حجم زیادی از تحقیقات در سال‌های اخیر به کاهش زمان نمونه‌برداری مدل‌های پخشی اختصاص داده شده‌است. برای دسته‌بندی این تحقیقات هم رویکردهای متفاوتی وجود دارد. به عنوان مثال می‌توان روش‌ها را بر اساس نیازمندی به آموزش مجدد[۳۹] و بدون نیاز به آموزش[۴۰] به دو دسته تقسیم کرد[۱۹].

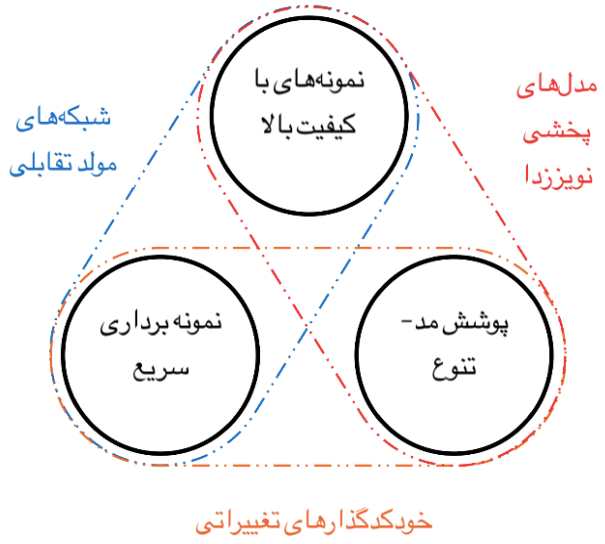
روش‌های مبتنی بر آموزش با تغییر قسمتی از مدل یا استفاده از رویکردهای تقطیر دانش[۴۱] [۸]سعی بر کاهش زمان فرآیند نمونه‌برداری و استفاده از شبکه‌های کوچک‌تر و سریع‌تر برای نویززدایی دارند. مشکل عمده‌ی این روش‌ها این است که برای فرآیند آموزش مجدد نیاز به هزینه‌ی زمانی و پردازشی زیادی دارند و این موضوع با توجه به گسترش ابعاد تصاویر و سنگین‌تر شدن شبکه‌های نویززدا نیز رو به افزایش است. در مقابل روش‌های بدون نیاز به آموزش، از یک شبکه‌ی از پیش آموزش داده‌شده

[38] Generative learning trilemma

[39] Training-based Methods

[40] Training-Free Methods

[41] Knowledge Distillation



شکل ۲-۷: مشکل سه‌گانه‌ی مدل‌های مولد
سه نیاز اساسی در توسعه‌ی مدل‌های مولد و حوزه‌ی عملکرد موثر هر مدل[۳۴].

[۴۲]استفاده می‌کنند و سعی می‌کنند تا با تغییر عناصر مختلف موجود در این مدل‌های از پیش آموزش داده شده، زمان نمونه برداری آنها را کاهش دهند.
رویکردهای بهبود سرعت را می‌توان از جنبه‌های دیگری هم دسته‌بندی کرد، یکی از این جنبه‌ها محل اثر رویکرد بهبود سرعت است. اساسا مشکل زمانبر بودن فرآیند نمونه‌برداری در مدل‌های پخشی از دو مساله نشات می‌گیرد. اول، تعداد گام‌های زیادی که برای نویززدایی نیاز است و دوم، سنگین بودن شبکه‌ی نویززدا که باعث زمانبر بودن محاسبات در هر گام به تنهایی می‌شود[۱۸]. بر همین اساس می‌توانیم رویکردهای افزایش سرعت را به دو دسته‌ی روش‌های کاهش تعداد گام و روش‌های کاهش محاسبات در هر گام تفکیک کنیم. در ادامه مختصرا به هر کدام می‌پردازیم.

۲-۲-۱ رویکردهای کاهش تعداد گام‌ها

رویکردهای موجود در این دسته تلاش می‌کنند تا با کاهش تعداد گام‌های نویززدایی فرآیند نمونه‌برداری را تسریع کنند. رویکردهای این دسته را مجدد و بر اساس دسته‌بندی کلی‌تر روش‌های کاهش زمان، می‌توان به دو دسته‌ی نیازمند یادگیری و بدون نیاز به یادگیری تقسیم کرد[۱۹]. یک دسته‌ی شاخص در این رویکردها روش‌های مبتنی بر تقطیر دانش هستند. روش‌های تقطیر دانش با انتقال دانش از مدل‌های سنگین و نیازمند نمونه‌برداری کامل معلم به مدل‌های سریع‌تر دانش‌آموز تعداد گام مورد نیاز برای فرآیند نمونه‌برداری را کاهش می‌دهند. بر اساس [۱۹] روش‌های تقطیر دانش نیز به سه دسته‌ی

[42] Pre-trained Network

تقطیر دانش مبتنی بر توزیع[۴۳]، تقطیر دانش مبتنی بر مسیر[۴۴] و تقطیر دانش تقابلی[۴۵] تقسیم می‌شوند. نقطه ضعف این مدل‌ها همانطور که در بحث رویکردهای نیازمند آموزش ذکر شد، فرآیند آموزش زمانبر و پر هزینه است و با توجه به اینکه عمدتا هم شبکه‌ی معلم و هم شبکه‌ی دانش‌آموز بزرگ هستند؛ منابع پردازشی قابل توجهی برای انجام آنها نیاز است. علاوه بر هزینه‌ی آموزشی زیاد در بعضی از روش‌های تقطیر مدل دانش‌آموز امکان درک فرمان‌های منفی[۴۶] و فرمان‌های ترکیبی را ندارد[۳۳].
رویکردهای دیگری که در دسته‌ی کاهش تعداد گام قرار می‌گیرند روش‌هایی هستند که با تغییر در قسمتی از فرآیند رو به جلو یا رو به عقب امکان پرش از روی چند گام را فراهم می‌کنند. از جمله‌ی این روش‌ها می‌توان به مدل‌های پخشی نویززدای ضمنی[۴۷] [۲۹] اشاره کرد. این روش با تغییر فرآیند رو به جلو به یک فرآیند غیر مارکفی، این امکان را فراهم می‌کند که در فرآیند رو به عقب با تعداد گام کمتر و بدون از دست دادن کیفیت، فرآیند نمونه برداری را انجام دهیم.
از بررسی مقالات مختلف موجود در این دسته می‌توان نتیجه‌گیری کرد که نقطه‌ی قوت اکثر روش‌هایی که در این دسته قرار می‌گیرند این است که بر مبنای یک تئوری ریاضیاتی بنا شده‌اند و در عین حال که انعطاف خوبی برای اعمال روی انواعی از معماری‌ها را دارند؛ می‌توانند به صورت عمود بر روش‌های کاهش محاسبات در هر گام بر مدل پخشی اعمال شوند تا حداکثر کاهش ممکن حاصل شود. از طرفی همانطور که در مثال‌های بالا هم مشاهده شد، عمده‌ی روش‌های شاخص در این دسته مبتنی بر آموزش هستند که نیاز به منابع پردازشی را به عنوان نقطه ضعف با خود دارند.

۲-۲-۲ رویکردهای کاهش محاسبات هر گام

روش‌های کاهش تعداد گام عمدتا ساختار شبکه‌ی نویز زدا را به صورت یک جعبه سیاه در نظر می‌گیرند. این در حالیست که چه در ساختار شبکه‌ی یو و چه در ساختار مبتنی بر مبدل قسمت‌های متعددی وجود دارند که امکان بهینه‌سازی و بهبود عملکرد دارند[۳۳]. عمده‌ی رویکردهایی که در این قسمت معرفی می‌کنیم بر مبنای تغییر در معماری، فشرده‌سازی شبکه و کاهش محاسبات تکراری می‌باشند و هدف تمامی این تغییرات کاهش حجم محاسبات انجام شده در هر گام می‌باشد.
رویکردهای کاهش‌محاسبات را می‌توان بر اساس محل اثر، هدف کلی و نیازمندی به آموزش به دسته‌های مختلفی تقسیم کرد. اما سه رویکرد شاخص، متاخر و عمدتا بدون نیاز به آموزش در این دسته وجود دارند که بررسی آنها برای درک سازوکار رویکردهای کاهش محاسبات مفید خواهد بود و این دسته‌ها

[43] Distribution-based KD
[44] Trajectory-based KD
[45] Adverserial KD
[46] Negative Prompts
[47] Denoising Diffusion Implicit Models(DDIM)

عبارتند از: گسسته‌سازی[۴۸]، کاهش توکن‌ها[۴۹] و استفاده از حافظه‌ی پنهان[۵۰] [۲۰].

۲-۲-۲-۱ گسسته‌سازی

گسسته‌سازی یک تکنیک برای کاهش دقت مقادیر عددی موجود در یک مدل است که معمولا با تبدیل اعداد ممیز شناور[۵۱] ۳۲ یا ۱۶ بیتی به بازنمایی‌های کم‌دقت‌تر مانند اعداد صحیح ۸ بیتی انجام می‌شود. رویکردهای گسسته‌سازی به دو دسته‌ی آموزش آگاه از گسسته‌سازی[۵۲] که در آنها فرآیند گسسته‌سازی همزمان با آموزش انجام می‌گیرد و گسسته‌سازی پس از آموزش[۵۳] تقسیم می‌شوند[۲۷].
در مدل‌های پخشی، فرآیند گسسته‌سازی باعث بهبود سرعت نمونه‌برداری و کاهش حجم شبکه‌ی نویززدا می‌شود و با توجه به فرآیند هزینه‌بر آموزش مدل‌های پخشی معمولا در این مدل‌ها از روش‌های پس از آموزش برای گسسته‌سازی شبکه استفاده می‌شود و این روش زمان و منابع محاسباتی کمتری نیاز دارد[۲۷]. همچنین با توجه به ماهیت چندگامی بودن مدل‌های پخشی عمده‌ی روش‌های گسسته‌سازی که پیش از این برای شبکه‌های عمیق تک گامی گسترش داده شده بودند، روی این مدل باعث افت کیفیت می‌شوند. از این رو مدل‌هایی مانند گسسته‌سازی پس از آموزش PTQ [۲۷]و گسسته‌سازی مدل پخشی[۵۴] [۱۵] ارائه شدند که با تمرکز بر چندگامی بودن مدل‌های پخشی و ماهیت متفاوت داده‌ی دریافت شده در هر گام به گسسته‌سازی شبکه‌ی نویززدا می‌پردازند.
به طور کلی فرآیند گسسته‌سازی بر وزن‌ها و خروجی فعالسازی[۵۵] مدل قابل اعمال است و در روش گسسته‌سازی مدل پخشی[۱۵] با فرمول زیر انجام می‌شود:

$$\hat{w} = \mathbf{s} \cdot \text{clip}\left(\text{round}\left(\frac{w}{\mathbf{s}}\right), \mathfrak{c}_{\min}, \mathfrak{c}_{\max}\right), \tag{۱۴-۲}$$

که در آن $\hat{w}$ مقدار پارامتر مدل بعد از گسسته‌سازی، s پارامتر تعیین مقیاس گسسته‌سازی، c_{min} و c_{max} بیانگر حد بالا و حد پایین تابع محدودسازی بازه[۵۶] و تابع $round$ نیز برای گرد کردن مقادیر به نزدیکترین مقدار یا گرد کردن به صورت تطبیقی تعریف می‌شوند.
در فرآیند گسسته‌سازی تمامی پارامترهای رابطه‌ی بالا به نحوی تعیین می‌شوند که خروجی مدل کمترین خطای ممکن را داشته باشد. برای وزن‌ها با توجه به ثابت بودن آنها در فاز پس از آموزش تعیین

[48] Quantization
[49] Token Reduction
[50] Caching
[51] Floating-point
[52] Quantization-aware Training
[53] Post-training Quantization
[54] Q-diffusion
[55] Activation
[56] Clipping function

پارامترهای گسسته‌سازی چندان دشوار نمی‌باشد. اما برای خروجی فعالسازی این موضوع با تعیین یک مجموعه داده‌ی کالیبراسیون[۵۷] و بهینه‌سازی خطای گسسته‌سازی بر روی آن تعیین می‌شود. در مساله‌ی گسسته‌سازی مدل‌های پخشی با توجه به ماهیت چندگامی بودن فرآیند نویززدایی تعیین مجموعه داده‌ی کالیبراسیون باید به گونه‌ای انجام شود که خطای گسسته‌سازی در هر گام زمانی نیز کمینه شود. در مقاله‌ی [۱۵] این چالش با در نظر گرفتن داده‌های نویزی با گام‌های زمانی مختلف در مجموعه داده برطرف شده است.

۲-۲-۲-۲ کاهش توکن‌ها

روش‌های کاهش توکن بر ورودی‌های مبدل‌ها اعمال می‌شوند و ایده‌ی کلی آنها این است که با توجه به اینکه هزینه‌ی محاسباتی مبدل‌ها بر حسب اندازه‌ی ورودی به صورت درجه دو است بنابراین کاهش اندازه‌ی ورودی، هزینه‌ی محاسباتی و تاخیر زمانی موجود در مبدل‌ها را کاهش می‌دهد و آنها را بهینه‌تر می‌سازد. تعریف توکن در مدل‌های مختلف متفاوت است. در مدل‌های پخشی پنهان با توجه به اینکه ساختار اصلی شبکه‌ی یو است، توکن به تک تک پیکسل‌های تصویر در فضای پنهان گفته می‌شود. برخلاف مدل پخشی پنهان، در مدل‌های مبتنی بر مبدل با توجه به اینکه کلیت شبکه‌ی نویززدا از تعداد زیادی کدگذار مبدل ایجاد شده است می‌تواند به صورت قطعه‌هایی از تصویر تعریف شود. در ادامه‌ی این پژوهش و در مرور کارهای معرفی شده کلمه‌ی توکن معادل تعریف آن در مدل پخشی پنهان در نظر گرفته می‌شود.

رویکردهای کاهش توکن در انواعی از مسائل از جمله مدل‌های زبانی، مبدل تصویری[۵۸] و انواع دیگری از مبدل‌ها استفاده شده است. به طور کلی رویکردهای کاهش توکن را می‌توان به سه دسته‌ی هرس توکن‌ها [۵۹]، ادغام توکن‌ها[۶۰] و روش‌های ترکیبی تقسیم کرد[۶][۱۰].

در هرس توکن‌ها یک مجموعه توکن به واسطه‌ی محاسبه‌ی یک تابع امتیاز بی اهمیت یا افزونه تشخیص داده می‌شوند و به طور کلی از محاسبات حذف می‌شوند. روش‌های هرس توکن میتوانند به دو صورت متغیر[۶۱] یا ثابت[۶۲] اعمال شوند. در حالت ثابت همواره نرخ مشخصی از توکن‌ها دور ریخته می‌شوند اما در رویکرد متغیر بسته به امتیاز محاسبه شده، تصویر ورودی و سایر شاخص‌ها میزان توکن‌های مناسب برای هرس تعیین می‌شود. تعیین میزان هرس مساله‌ی مهمی است و هرس بیش از حد باعث از دست رفتن اطلاعات و خطا در مدل می‌شود[۱۰].

روش‌های ادغام توکن‌ها به جای حذف توکن و کاهش اندازه‌ی ورودی مبدل از این طریق، توکن‌های مشابه را پیدا می‌کنند و آنها را با استفاده از یک روش ادغام ترکیب می‌کنند. این رویکردها نیز به

[57] Calibration Dataset
[58] Vision Transformer(VIT)
[59] Token Pruning
[60] Token Merging
[61] Dynamic
[62] Static

دو دسته‌ی ادغام سخت[۶۳] و ادغام نرم[۶۴] تقسیم می‌شوند. در روش‌های سخت معمولا از رویکردهای خوشه‌بندی برای تعیین توکن‌های قابل ادغام استفاده می‌شود در حالیکه در روش‌های نرم معمولا از دوبخشی کردن توکن‌ها و محاسبه‌ی معیار شباهت استفاده می‌شود[۶]. در روش‌های ترکیبی نیز عمدتا از ترکیب ایده‌های هرس و ادغام توکن‌ها برای کاهش توکن استفاده می‌شود.
در معماری $Stable\ Diffusion\ v1.5$ ۱۵ بلوک $BasicTransformer$ در سطوح مختلف و بلوک‌های کدگذار و کدگشا استفاده شده است[۲] و با توجه به اینکه شبکه‌ی یوی نویززدا از این مبدل و بلوک‌های باقی‌مانده‌ای تشکیل شده است، کاهش حجم محاسبات قابل انجام در یک بلوک مبدل می‌تواند به عنوان یکی از رویکردهای کاهش حجم محاسبات مدل در هر گام در نظر گرفته شود[۲۳]. در ادامه دو رویکرد پیاده‌سازی شده بر روی مدل‌های پخشی را معرفی می‌کنیم.

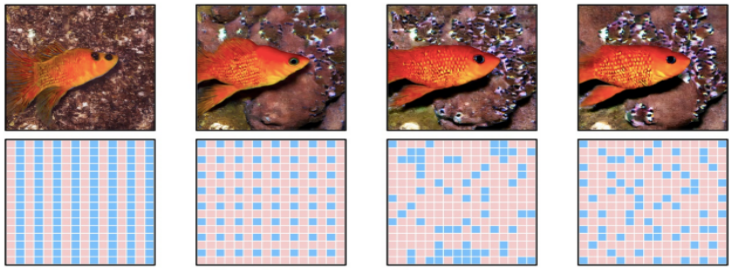
۲-۲-۲-۱ روش ادغام توکن‌ها

اکثر تصاویر از جمله آنهایی که در یک مدل پخشی تولید می‌شوند دارای افزونگی هستند و از آنجاییکه که مبدل تک تک توکن‌ها را پردازش می‌کند، حجم قابل توجهی از محاسبات غیر ضروری به مدل تحمیل می‌شود. روش ادغام توکن‌ها[۶۵] [۲] یک روش برای کاهش توکن‌های ورودی مبدل است. این روش ابتدا بر روی مبدل بینایی و سپس بر روی مدل پخشی پنهان $Stable\ Diffusion\ v1.5$ پیاده‌سازی و اجرا شده است.

این روش با تقسیم توکن‌ها به دو دسته‌ی مبدا [۶۶]و مقصد[۶۷] و ساخت یک گراف دوبخشی از آنها روش ادغام را پیاده‌سازی می‌کند. برای دسته‌بندی هر توکن به عنوان مبدا یا مقصد رویکردهای مختلفی وجود دارد. شکل ۲-۸ انواعی از این حالات را نشان می‌دهد. انتخاب تصادفی مبدا و مقصد باعث مشکلات زیادی می‌شود از جمله اینکه نواحی از تصویر می‌توانند هیچ توکن مقصدی نداشته باشند در نتیجه برای ادغام انتخاب نشوند در حالیکه قسمتی از پس زمینه یا نواحی کمتر متغیر بوده اند. بنابراین رویکرد مناسب‌تر در نظر گرفتن یک پنجره‌ی ۲ در ۲ در تصویر و انتخاب یکی از عناصر آن به صورت تصادفی برای مقصد و بقیه به عنوان مبدا است.

بعد از ساخت گراف دوبخشی مطابق شکل ۲-۹ از توکن‌های مبدا و مقصد یک ماتریس مشابهت کسینوسی[۶۸] بین هر توکن مبدا و توکن‌های مقصد محاسبه می‌شود و بر این اساس r درصد توکن‌ها با بیشترین مشابهت با یکدیگر ادغام می‌شوند. ادغام نیز به صورت میانگین‌گیری میان توکن‌های مورد

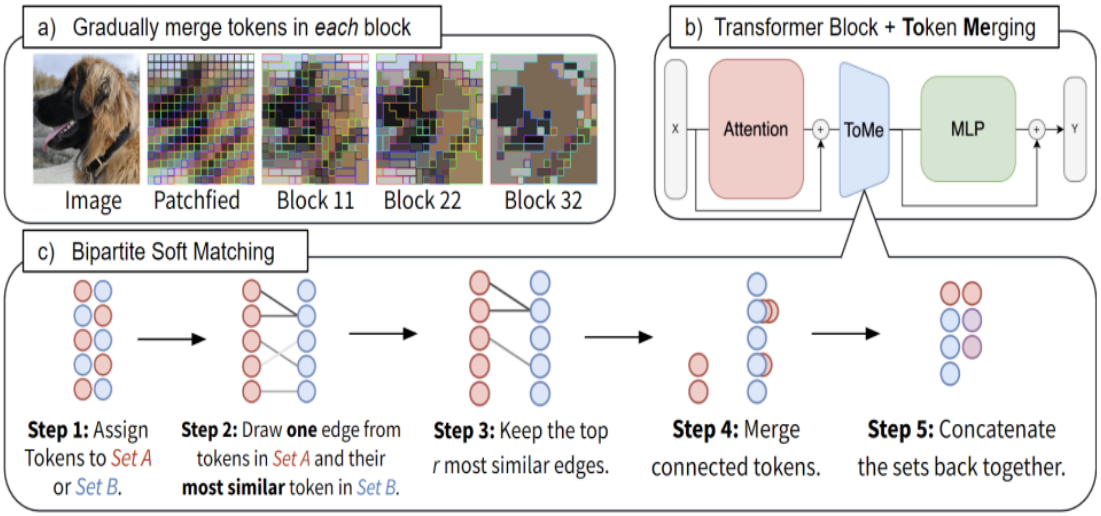
[63] Hard-merging
[64] Soft-merging
[65] ToMe
[66] Source
[67] Destination
[68] Cosin Similarity



شکل ۲-۸: رویکردهای انتخاب توکن‌های مبدا و مقصد در ادغام توکن‌ها

توکن‌های مبدا با رنگ قرمز و توکن‌های مقصد با رنگ آبی مشخص شده‌اند. روش a انتخاب یکی در میان توکن‌ها به عنوان مبدا و مقصد را نشان می‌دهد. روش b انتخاب توکن با گام پرش ۲ را نشان می‌دهد که می‌توانیم به صورت انتخاب توکن بالا سمت راست در یک پنجره‌ی ۲ در ۲ نیز آن را تعریف کنیم. روش c انتخاب ۲۵ درصد از توکن‌ها به صورت تصادفی را نشان می‌دهد و روش d نیز انتخاب تصادفی یک توکن در یک پنجره‌ی ۲ در ۲ را نشان می‌دهد[۲].

ادغام انجام می‌شود. فرآیند رفع ادغام[۶۹] نیز به صورت جایگذاری توکن ادغام شده در تک تک توکن‌های درگیر در ادغام انجام می‌شود.



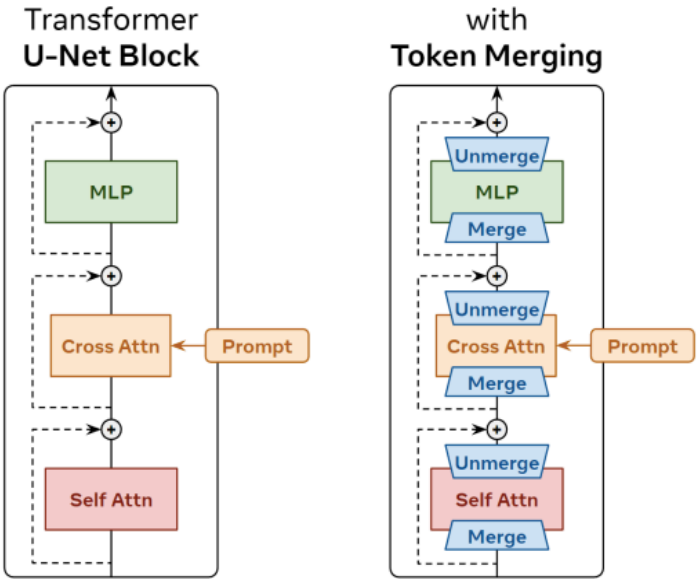
شکل ۲-۹: روال کار ساخت گراف دوبخشی و ادغام توکن‌ها

مجموعه‌ی A و B در این تصویر همان توکن‌های مبدا و مقصد هستند و این روش مراحل ساخت گراف دوبخشی و انتخاب توکن‌هایی که باهم ادغام می‌شوند را نشان می‌دهد[۱].

شکل ۲-۱۰ محل قرارگیری بلوک‌های ادغام و رفع ادغام را نشان می‌دهد. این بلوک‌ها می‌توانند در

[69] Unmerging

اول و آخر بلوک توجه به خود یا توجه متقابل یا بلوک تماما متصل مبدل‌های موجود در مدل پخشی قرار گیرند. این رویکرد باعث افزایش سرعت در عین حفظ کیفیت می‌شود.



شکل ۲-۱۰: محل قرارگیری بلوک‌های ادغام و رفع ادغام در بلوک مبدل در مدل‌های پخشی

همانطور که در شکل مشاهده می‌کنید این بلوک در ابتدا و انتهای بلوک توجه به خود، توجه متقابل یا بلوک تماما متصل مبدل‌های موجود در مدل پخشی قرار می‌گیرد[۲].

۲-۲-۲-۲-۲ روش همجوشی توکن‌ها

روش همجوشی توکن‌ها[۷۰] [۱۰] پس از روش ادغام توکن‌ها و برای مبدل بینایی ارائه شده و سپس روی مدل‌های پخشی نیز پیاده‌سازی و ارزیابی شد. این روش یک روش ترکیبی از مزایای ادغام توکن‌ها و هرس آنهاست. روش هرس توکن‌ها معمولا نیاز به تعریف یک تابع تعیین اهمیت یا امتیازدهی به توکن‌ها دارد و در عین حال حذف بیش از حد در این روش باعث از بین رفتن اطلاعات می‌شود. در مقابل روش ادغام توکن‌ها یک روش بدون نیاز به امتیازدهی است اما در حالتی که ادغام با میانگین‌گیری انجام می‌شود، باعث تغییر در توزیع داده‌ها می‌شود. از این رو در روش همجوشی توکن‌ها به دنبال یک ایده‌ی بینابینی هستیم.

روش همجوشی توکن‌ها بر اساس معیار خطی بودن تابعی[۷۱] تعریف می‌شود. این معیار بررسی می‌کند که یک مدل تا چه اندازه نسبت به درونیابی ورودی‌هایش خطی عمل می‌کند. در شبکه‌های عصبی یک لایه به تنهایی چندان از معیار خطی بودن تابعی برخوردار نیست اما افزایش عمق و افزودن لایه مخصوصا در مبدل‌های بینایی این معیار را تقویت می‌کند. روش همجوشی از این معیار برای تعیین اینکه در کجا

[70] Token Fusion

[71] Functional Linearity

باید از هرس توکن‌ها استفاده شود و در کجا باید از ادغام توکن‌ها استفاده شود، بهره می‌گیرد.
در لایه‌هایی که خطی بودن تابعی کم است، استفاده از هرس توکن‌ها بهتر است چرا که ادغام توکن‌ها با میانگین یا هر عملگر دیگری باعث انحراف از توزیع یاد گرفته شده توسط مدل و تغییر در توزیع داده می‌شود؛ و بنابراین استفاده از هرس توکن‌ها که تنها توکن‌های کم‌اهمیت را حذف می‌کند و درونیابی صورت نمی‌گیرد بهتر است.
در مقابل، در لایه‌هایی که خطی بودن تابعی زیادی دارند، مدل نسبت به ورودی‌های درونیابی شده نتیجه‌ی قابل پیش‌بینی‌تری دارد و در نتیجه استفاده از روش‌های ادغام توکن‌ها و مزایای آنها در حفظ اطلاعات می‌تواند بهتر باشد. بر این اساس روش همجوشی توکن‌ها با تحلیل میزان برقراری خطی بودن تابعی در لایه‌های اولیه از هرس توکن‌ها و در لایه‌های عمیق‌تر از ادغام توکن‌ها استفاده می‌کند.
همچنین با توجه به اینکه میانگین‌گیری باعث کاهش نرم[۷۲] ویژگی‌ها می‌شود، و این کاهش نرم می‌تواند منجر به تغییر در توزیع شود، این روش به جای میانگین‌گیری در زمان ادغام از معیار ترکیبی درونیابی خطی حداکثر نرم[۷۳] $MLERP$ استفاده می‌کند که فرمول آن به صورت زیر است[۱۰]:

$$\bar{x} = \frac{1}{K}\sum_{k=1}^{K} x_k \tag{۱۵-۲}$$

$$MLERP(\{(x_1, \ldots, x_K)\}) = \frac{\bar{x}}{\|\bar{x}\|} \times \max_{k=1,\ldots,K} \|x_k\| \tag{۱۶-۲}$$

نتایج پیاده‌سازی این روش بر روی مدل‌های پخشی نشان می‌دهد که این روش نسبت به روش ادغام توکن‌ها کاهش زمانی بیشتری دارد.

۲-۲-۲-۳ استفاده از حافظه‌ی پنهان

همانطور که پیش‌تر هم اشاره شد روش‌های کاهش گام عمدتا شبکه‌ی نویززدا را به عنوان یک جعبه‌ی سیاه در نظر می‌گرفتند و بررسی عملکرد بلوک‌های مختلف آن و تاثیر آنها بر سرعت و کیفیت تصویر از جمله مواردی بود که کمتر مورد توجه قرار می‌گرفت[۳۳]. اما ظهور مدل استفاده از حافظه‌ی پنهان عمیق[۷۴] [۱۸] باعث ایجاد دسته‌ای از روش‌های استفاده از حافظه‌ی پنهان شد که با بررسی افزونگی[۷۵] موجود در قسمت‌های مختلف مدل‌های پخشی، این رویکرد را در جهت کاهش حجم محاسبات و کاهش زمان نمونه‌برداری استفاده کردند.

[72] Norm
[73] Maximum Norm Linear Interpolation
[74] Deep Cache
[75] Redundancy

در روش استفاده از حافظه پنهان با اتکا به این موضوع که محاسبات در بلوک‌های مختلف و در گام‌های زمانی مختلف دارای افزونگی هستند، قسمتی از محاسبات در بلوک مربوطه انجام نمی‌شود و این محاسبات از بلوک قبلی یا گام زمانی قبلی که در حافظه‌ی پنهان ذخیره شده است، جایگذاری می‌شود. روش استفاده از حافظه‌ی نهان بر ویژگی‌های سطح بالای شبکه‌ی یو، کدگشای شبکه‌ی یو، بلوک توجه متقابل و بر تغییرات محاسبه اعمال شده است که در هر یک از مثال‌های زیر یکی از این حالات را ارائه می‌کنیم.

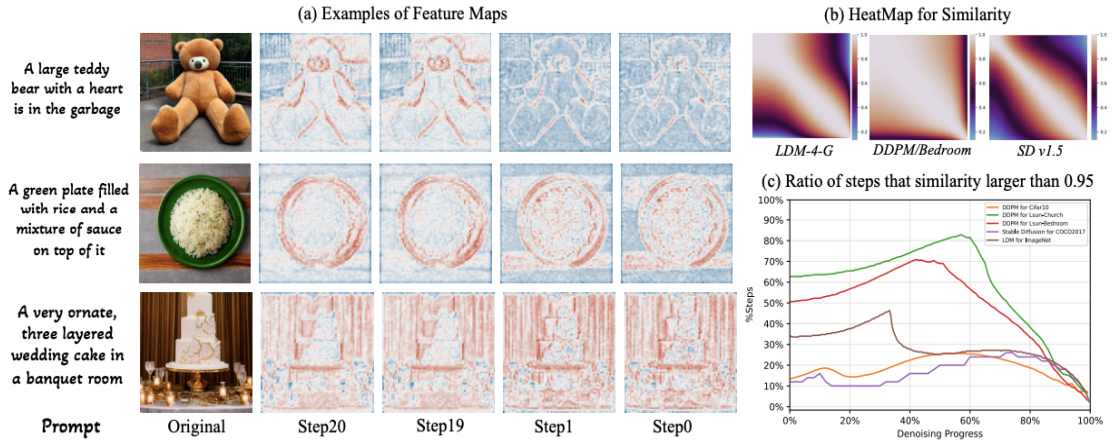
۲-۲-۲-۳-۱ روش استفاده از حافظه‌ی پنهان عمیق

روش استفاده از حافظه‌ی پنهان عمیق[۱۸] بر مبنای یک مشاهده‌ی تجربی توسعه داده شده است. این مشاهده بیان می‌کند که گام‌های مجاور در فرآیند نویززدایی افزونگی زمانی زیادی را در ویژگی‌های سطح بالا نشان می‌دهند. ویژگی‌های سطح بالا به مجموعه ویژگی‌های حاصل از عبور تصویر از بلوک‌های متوالی کدگذار در شبکه‌ی یو می‌گوییم و منظور از افزونگی زمانی این است که اگر خروجی این بلوک‌ها در دو گام متوالی را مشاهده کنیم تغییرات چشم‌گیری در آنها دیده نمی‌شود. این موضوع را می‌توان به این شکل هم صورت‌بندی کرد که تغییرات در گام‌های متوالی بسیار تدریجی و آهسته است.
شکل ۲-۱۱ این مشاهدات تجربی را نشان می‌دهد. قسمت a خروجی بلوک نمونه‌افزای[۷۶] U_2 را در دو گام زمانی متوالی نشان می‌دهد که شامل تغییرات خیلی ملموسی نیستند. قسمت b نمودار نقشه حرارتی[۷۷] مربوط به مشابهت را در سه مدل پخشی مختلف نشان می‌دهد و همانطور که مشاهده می‌کنید در نزدیک قطر اصلی یعنی در گام‌های نسبتا مجاور این عدد شباهت بسیار نزدیک است که موید همان مشاهده‌ی مطرح شده در بند قبل است و در نهایت قسمت c نیز درصد تعداد گام‌هایی که با گام مورد نظر شباهتی بیشتر از ۰.۹۵ دارند را نشان می‌دهد. دقت کنید که تمام نمودارها برای خروجی‌های بلوک نمونه‌افزای U_2 است که در شکل ۲-۱۲ مکان آن را در مدل $Stable\ Diffusion$ می‌توانید مشاهده کنید.

شکل ۲-۱۲ کلیت این روش را نشان می‌دهد. در این شکل بلوک‌های D_i بلوک‌های زیر نمونه‌گیری[۷۸] در کدگذار شبکه‌ی یو و بلوک‌های U_i بلوک‌های نمونه افزای کدگشای شبکه‌ی یو هستند. این رویکرد بعد از محاسبه‌ی خروجی U_2 برای یک گام $t+1$ و ذخیره‌ی آن در حافظه‌ی پنهان، در گام t به جای محاسبه‌ی ویژگی‌های سطح بالای بلوک‌های میانی آن را از حافظه‌ی نهان فراخوانی می‌کند و به کمک اتصال پرشی و بلوک U_1 خروجی این گام را محاسبه می‌کند.

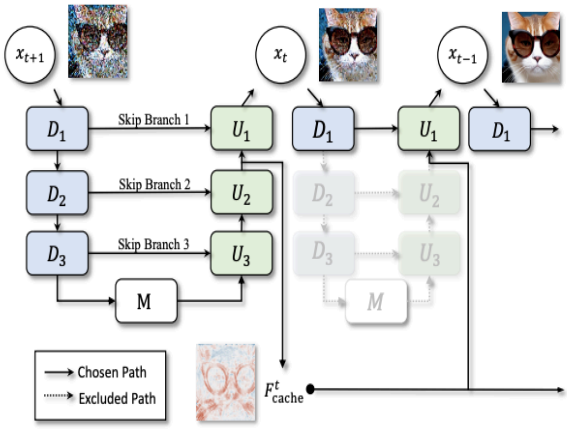
نکته‌ی دیگری که حائز اهمیت است این است که در چه گام‌هایی محاسبه‌ی مجدد انجام شود و در چه گام‌هایی از حافظه‌ی پنهان فراخوانی شود. در این روش دو رویکرد مطرح می‌شود، رویکرد اول که رویکرد یکنواخت[۷۹] نام‌گذاری شده است، بیان می‌کند که هر N گام یک بار نیاز است که محاسبات

[76] Upsampling
[77] Heatmap
[78] Downsampling
[79] Uniform



شکل ۲-۱۱: مشاهدات تجربی مبنی بر وجود افزونگی زمانی در ویژگی‌های سطح بالا در مدل‌های پخشی

قسمت a نقشه‌ی ویژگی دو مجموعه گام متوالی را در بلوک U_2 نشان می‌دهد. قسمت b نقشه‌ی حرارتی ویژگی‌های مشابهت را در گام‌ها زمانی نشان می‌دهد. قسمت c هم نمودار درصد گام‌هایی که با گام مد نظر مشابهتی بیشتر از ۰.۹۵ را دارند نشان می‌دهد[۱۸].



شکل ۲-۱۲: کلیت روش استفاده از حافظه‌ی پنهان عمیق

شکل کلیت این روش را در دو گام $t+1$ و t نشان می‌دهد. خروجی بلوک U_2 در حافظه‌ی پنهان ذخیره شده و در گام بعدی به جای محاسبه‌ی ویژگی‌های سطح بالا از حافظه‌ی پنهان خروجی گام قبل استفاده شده است[۱۸].

مجدد انجام شوند و مجموعه‌ی $\mathcal{I}$ که گام‌های محاسبه است را به صورت زیر تعریف می‌کند:

$$\mathcal{I} = \{x \in \mathbb{N} \,|\, x = iN,\ 0 \leq i < k\} \tag{۲-۱۷}$$

که $\mathbb{N}$ مجموعه‌ی تمامی گام‌ها از صفر تا T است و k نیز برابر با $\frac{T}{N}$ است. در حالت یکنواخت ما به

صورت ضمنی این فرض را داریم که در هر N گام متوالی ویژگی‌های سطح بالا تغییر بسیار ناچیزی دارند، در حالیکه این مخصوصا در N های بزرگ فرض کاملا اشتباهی است. با توجه به شکل ۲-۱۱ قسمت c بعد از ۴۰ درصد فرآیند نویز زدایی شباهت زمانی بین ویژگی‌ها کاهش پیدا می‌کند از این رو استفاده از یک رویکرد غیر یکنواخت[۸۰] برای مشخص کردن مجموعه‌ی $\mathcal{I}$ اهمیت بسیاری دارد. مجموعه‌ی غیر یکنواخت $\mathcal{I}$ به صورت زیر تعریف می‌شود:

$$\mathcal{L} = \left\{ l_i \mid l_i \in linearspace\left((-c)^{\frac{1}{p}}, (T-c)^{\frac{1}{p}}, k \right) \right\} \tag{۲-۱۸}$$

$$\mathcal{I} = uniqueint\left(\{ i_k \mid i_k = (l_k)^p + c,\ where\ l_k \in \mathcal{L} \} \right) \tag{۲-۱۹}$$

که c یک پارامتر برای تعیین گام زمانی مرکزی می‌باشد و تابع $linearspace(n, s, e)$ تعداد n نقطه با فاصله‌ی مساوی بین s و e در نظر می‌گیرد. همچنین تابع $uniqueint$ هم اطمینان حاصل می‌کند که گام‌های زمانی صحیح و بدون تکرار باشند. در این رابطه فرکانس‌های زمان‌های محاسبه به صورت دوجمله‌ای و از یک گام زمانی اولیه محاسبه می‌شوند این روش بر روی $Stable\ Diffusion\ v1.5$ با ضریب ۱۱.۲ کاهش سرعت می‌دهد و با اندکی افزایش معیار FID همراه است به این معنا که کیفیت تصویر اندکی کاهش پیدا می‌کند.

۲-۲-۲-۳-۲ مدل پخشی سریع‌تر

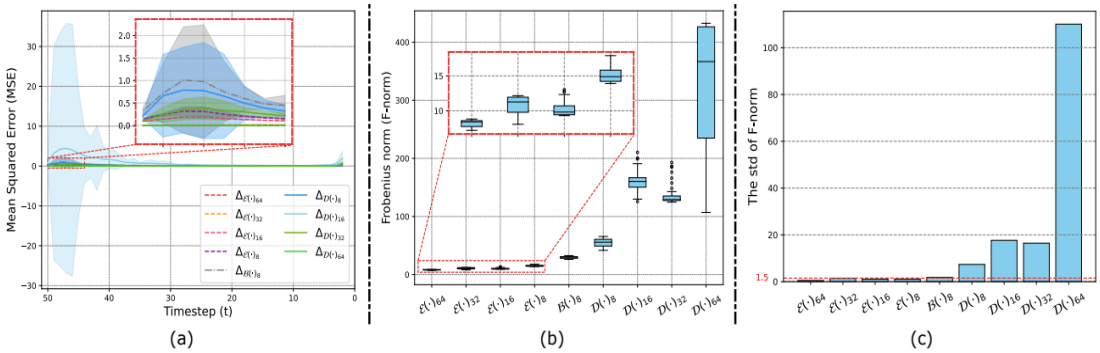
روش مدل پخشی سریعتر[۸۱] [۱۴] سعی دارد تا با ذخیره‌ی خروجی کدگذار شبکه‌ی یوی مدل پخشی، فرآیند نمونه‌برداری مدل‌های پخشی را موازی‌سازی کند. این موضوع مجددا از این ایده سرچشمه می‌گیرد که به صورت تجربی قابل مشاهده است که ویژگی‌های خروجی کدگذار شبکه‌ی یوی یک مدل پخشی تغییرات بسیار جزئی دارد و در مقابل خروجی کدگشا به طور قابل توجهی تغییر می‌کند.
در این روش برای بررسی میزان تغییرات ویژگی‌های شبکه‌ی یو، بلوک‌های این شبکه را به سه دسته‌ی بلوک‌های کدگذار که شامل $\mathcal{E}(.)_8, \mathcal{E}(.)_{16}, \mathcal{E}(.)_{32}, \mathcal{E}(.)_{64}$ و بلوک میانی یا گلوگاه[۸۲] $\mathcal{B}(.)$ و بلوک‌های کدگشا که شامل $\mathcal{D}(.)_8, \mathcal{D}(.)_{16}, \mathcal{D}(.)_{32}, \mathcal{D}(.)_{64}$ است تقسیم می‌کنند و سپس با محاسبه‌ی نرم[۸۳] تفاضل هر بلوک در دو زمان متوالی به فرم زیر معیاری برای شباهت بلوک‌ها در زمان‌های متوالی تعریف می‌کنند.

$$\Delta\mathcal{E}(\cdot)_s = \frac{1}{d \times s^2} \left\| \mathcal{E}(z_t, c, t)_s - \mathcal{E}(z_{t-1}, c, t-1)_s \right\|_2^2, \tag{۲-۲۰}$$

[80] Non-uniform
[81] Faster Diffusion
[82] Bottleneck
[83] Norm

که در این رابطه d نمایانگر تعداد کانال ورودی و s تعیین کننده‌ی شماره‌ی بلوک عدد محاسبه شده است.

شکل ۲-۱۳ نمودارهای تجربی به دست آمده از محاسبه‌ی تفاضل بالا در بلوک‌ها و گام‌های زمانی متفاوت را نشان می‌دهد. در قسمت a روند این تفاضل در گام‌های زمانی متوالی آمده است و نکته‌ی حائز اهمیت مقیاس واریانس تغییر این تفاصل در کدگذار و کدگشا است. در کد گذار واریانس این ترند بین ۰.۰۵ و ۰.۴ تغییر می‌کند و این در حالی است که در کدگشا این عدد بین ۵ تا ۳۰ تغییر می‌کند که نشان دهنده‌ی وجود تغییر بسیار در کدگشا و تغییر بسیار کم در کدگذار است. در قسمت b برای ویژگی‌های خروجی هر بلوک ابتدا میانگین آنها در راستای کانال محاسبه شده و سپس نرم فربنیوس[۸۴] هر ویژگی محاسبه شده و این عدد برای گام‌های زمانی مختلف محاسبه شده است. همانطور که می‌بینید ارتفاع نمودار جعبه‌ای در کدگذار بسیار ناچیز است و این به این معناست که تغییرات در کدگذار در گام‌های زمانی متفاوت ناچیز است. در قسمت c نیز مشابه قسمت b نرم فربینیوس ویژگی‌ها محاسبه شده است و در این نمودار انحراف معیار آن برای برای بلوک‌های مختلف مشاهده می‌کنیم. این شکل نیز موید این موضوع است که تغییرات در کدگذار بسیار ناچیزتر از کدگشا هستند.



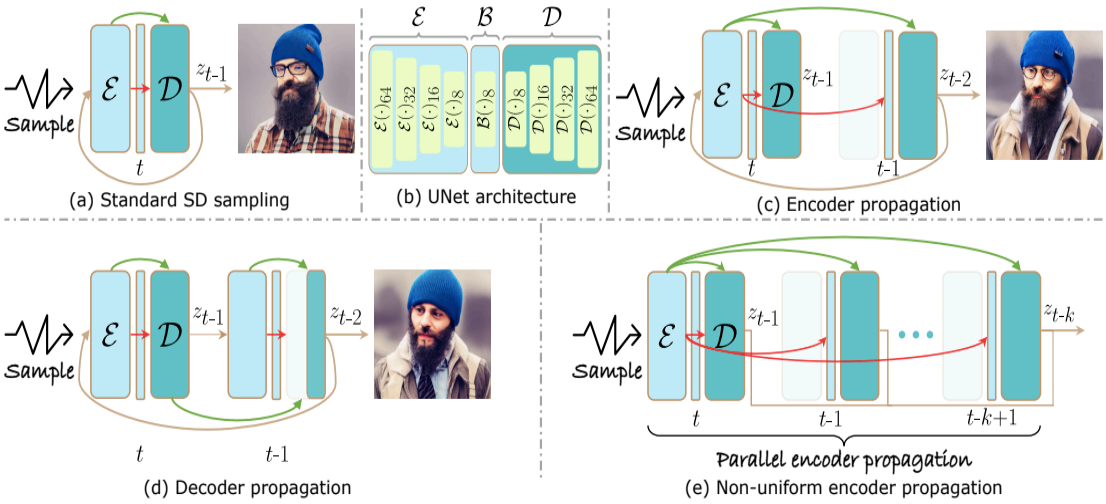
شکل ۲-۱۳: مشاهدات تجربی مبتنی بر تغییرات ناچیز کدگذار شبکه‌ی یوی مدل پخشی

هر سه بخش این تصویر معیارهایی برای بازنمایی میزان تغییرات ویژگی‌های خروجی کدگذار در مقابل کدگشا و بلوک میانی هستند. همانطور که مشاهده می‌شود ویژگی‌های کدگذار تغییرات جزئی‌تری نسبت به کدگشا دارند[۱۴].

بر مبنای این مشاهدات ایده‌ی اصلی روش مدل پخشی سریع‌تر استفاده از حافظه‌ی پنهان برای نگه‌داری ویژگی‌های کدگذار در بعضی از گام‌ها و محاسبه‌ی آن در سایر گام‌ها است. این ایده که انتشار کدگذار[۸۵] نام دارد در شکل ۲-۱۴ آمده است. همانطور که در شکل هم قابل مشاهده است، ذخیره‌ی کدگذار در حافظه‌ی پنهان امکان موازی‌سازی فرآیند نویززدایی را هم فراهم می‌آورد. استفاده از این رویکرد برای مدل $Stable\ Diffusion$ با ۴۱ درصد افزایش سرعت در عین حفظ کیفیت همراه است.

[84] Frobenius Norm

[85] Encoder Propagation



شکل ۲-۱۴: کلیت مدل پخشی سریع‌تر

قسمت a فرآیند نمونه‌برداری عادی در یک مدل پخشی را نشان می‌دهد. قسمت b ساختار یک شبکه‌ی یوی به کار رفته در مدل پخشی را نشان می‌دهد. قسمت c فرآیند انتشار کدگذار برای دو گام متوالی را نشان می‌دهد. قسمت d فرآیند انتشار کدگشا را نشان می‌دهد که تصویر تولید شده قسمت‌هایی از فرمان داده شده را به خوبی اجرا نمی‌کند. قسمت e هم فرآیند موازی‌سازی انتشار کدگذار را نشان می‌دهد[۱۴].

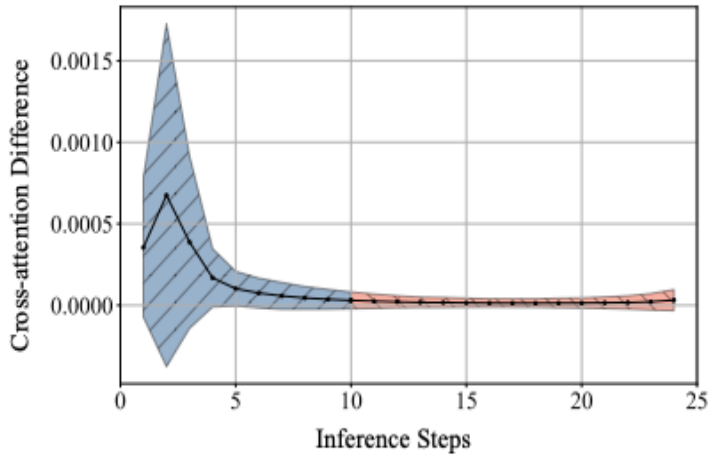
۲-۲-۳-۳ رویکرد گیت T

رویکرد گیت T[۸۶] [۱۷] با تمرکز بر بلوک توجه متقابل از حافظه‌ی پنهان استفاده می‌کند. ایده‌ی کلی این روش این است که فرآیند نویززدایی دو قسمت دارد. قسمت اول که فاز برنامه‌ریزی معنایی است[۸۷]، شبکه به فرمان[۸۸] توجه می‌کند تا با توجه به آن کلیات معنایی تصویر را بسازد و بعد در قسمت دوم که به آن افزایش کیفیت[۸۹] می‌گوییم، کلیات معنایی را بهبود می‌دهیم و به آنها جزئیات اضافه می کنیم. از آنجایی که مسئولیت روش بلوک توجه متقابل درگیر کردن فرمان در فرآیند تولید نمونه است، ایده‌ی این روش این است که پس از ساخت کلیت معنایی دیگر نیازی به محاسبه‌ی دقیق بلوک توجه متقابل نداریم و می‌توانیم آن را از حافظه‌ی پنهان بازیابی کنیم.

در این مقاله، دو آزمایش برای بررسی نقش بلوک توجه متقابل انجام شده است. در آزمایش اول مشابه دو روش قبل، تفاضل خروجی بلوک توجه متقابل در گام‌های متوالی برای تعدادی تصویر محاسبه شده است. شکل ۲-۱۵ خروجی این آزمایش را نشان می‌دهد. همانطور که مشاهده می‌کنید، تفاضل خروجی بلوک توجه متقابل در چند گام اول نسبت به گام‌های بعدی بیشتر است و این تفاضل در گام ۵ تا ۱۰ به همگرایی می‌رسد و بعد از آن تقریبا ثابت می‌ماند.

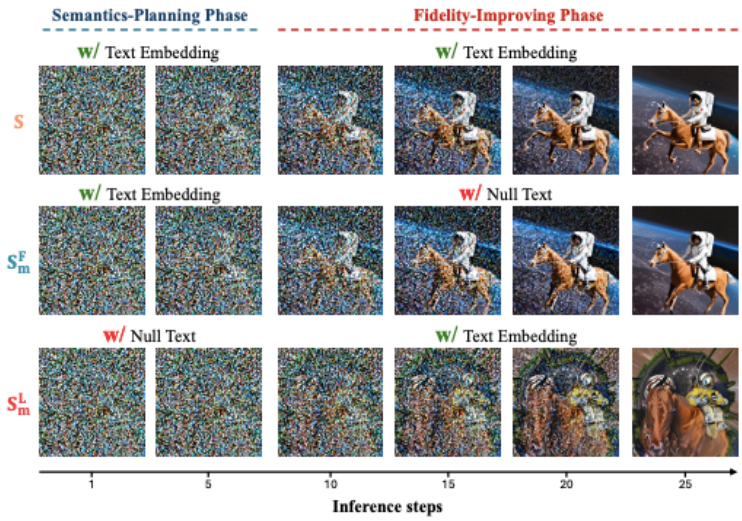
آزمایش دوم برای اثبات نقش بلوک توجه متقابل و نیز برای جداسازی فاز برنامه‌ریزی معنایی از فاز

[86] TGate
[87] Semantic Planning
[88] Prompt
[89] Fidelity Improving



شکل ۲-۱۵: نمودار تفاضل تغییرات بلوک توجه متقابل در گام‌های زمانی متوالی

همانطور که نمودار نشان می‌دهد تغییرات در گام‌های ابتدایی بسیار زیاد است و در حول و حوش گام ۵ تا ۱۰ همگرا می‌شود[۱۷].



شکل ۲-۱۶: تقسیم فرآیند نویززدایی به دو بخش برنامه‌ریزی معنایی و افزایش کیفیت

این شکل حالت‌های مختلف حذف فرمان را نشان می‌دهد. همانطور که مشاهده می‌کنید حذف فرمان در گام‌های ابتدایی باعث کاهش چشم‌گیر کیفیت تصویر می‌شود در حالیکه در حالت برعکس کیفیت تصویر تغییر چندانی نمی‌کند[۱۷].

افزایش کیفیت ارائه شده است. این آزمایش به این صورت است که به ازای یک m بین ۰ تا ۲۵ که گام‌های زمانی نویززدایی هستند مجموعه‌ی گام ها را به دو بخش تقسیم می‌کنیم و در یک بخش فرمان را به کمک بلوک توجه متقابل اعمال می‌کنیم و در بخش دوم فرمان را اعمال نمی‌کنیم. سپس تفاوت کیفیت تصویر تولید شده در این حالت با حالت طبیعی مدل پخشی را می‌سنجیم، اگر در چنین حالتی تفاوت ناچیز باشد می‌توانیم نتیجه بگیریم که از گام‌های بعد از m نیازی به اعمال فرمان و محاسبه‌ی

توجه متقابل نیست. شکل ۲-۱۶ این آزمایش را نشان می‌دهد.
خروجی این رویکرد به ازای m های مختلف بین ۱۰ تا ۵۰ درصد امکان بهبود سرعت مدل پخشی با حفظ کیفیت را دارد.

۲-۳ جمع‌بندی

در این فصل با شروع از تعریف کلی مدل‌های پخشی و ارائه‌ی جزئیات در خصوص مدل‌های پخشی نویززدای احتمالاتی و بیان این موضوع که نقطه ضعف اساسی این مدل‌ها سرعت آهسته‌ی فرآیند تولید نمونه‌ی آنها است به بررسی انواعی از روش‌های افزایش سرعت و دسته‌بندی‌های آنها پرداختیم. همچنین با توجه به اینکه راهکار اصلی بیان شده در این پژوهش ترکیبی از روش‌های مبتنی بر حافظه‌ی پنهان و ادغام توکن‌ها است، تلاش کردیم تا رویکردهای مطرح در این دو دسته را به طور دقیق‌تری مورد بررسی و مرور قرار دهیم.

فصل سوم

روش پیشنهادی

در این فصل به معرفی روش پیشنهادی می‌پردازیم. برای ارائه‌ی روش پیشنهادی ابتدا به بررسی دقیق‌تر روش ادغام توکن‌ها که در فصل قبل معرفی شد می‌پردازیم و سپس با بیان نقاط ضعف و قابل بهبود، روش خود را توضیح می‌دهیم. روش پیشنهادی از ترکیب دو ایده‌ی تطبیقی ساختن فرآیند ادغام و استفاده از حافظه‌ی پنهان در فرآیند ادغام می‌باشد که در ادامه هر یک را معرفی می‌کنیم.

۳-۱ روش پایه

روش پایه‌ی ما در این پژوهش روش ادغام توکن‌ها[1] [۲] است. این روش ابتدا برای مبدل بینایی و سپس برای مدل‌های پخشی ارائه شد. در این قسمت ابتدا سعی می‌کنیم جزئیات مختلف این روش را بیشتر باز کنیم و به بررسی انتخاب‌های نویسندگان مقاله بپردازیم.
پیش‌تر گفته شد که روش ادغام توکن‌ها بر روی ساختارهای مبدل موجود در مدل پخشی پیاده‌سازی می‌شود و با توجه به شکل ۲-۱۰ می‌توانیم ببینیم که در هر بلوک مبدل نیز بر سه زیربلوک توجه به خود، توجه متقابل و تماما متصل قابل اعمال است. اما سوال مهم‌تر این است که به کدامیک از سطوح شبکه‌ی یو قابل اعمال است. شبکه‌ی یو دارای سه لایه‌ی زیرنمونه‌گیری، یک لایه‌ی میانی و سه لایه‌ی نمونه‌افزایی است و در همه‌ی این لایه‌ها بلوک مبدل وجود دارد.
در این روش، لایه‌هایی که مورد فرآیند ادغام قرار می‌گیرند با ابرپارامتری[2] به نام $max\ downsample$ تعیین می‌شود. این پارامتر از تقسیم اندازه‌ی توکن‌های اصلی به توکن‌های ورودی در هر لایه محاسبه می‌شود و می‌تواند مقادیر ۱، ۲، ۴ و ۸ را به خود بگیرد. با انتخاب عدد ۸ ادغام توکن بر روی هر بلوک مبدل موجود در شبکه‌ی یو اعمال می‌شود. جدول ۳-۱ نتایج اعمال اعداد مختلف را نشان می‌دهد و بر اساس این مشاهده دو موضوع قابل بیان است اول اینکه افزایش ادغام در لایه‌هایی که ورودی کمتری دارند می‌تواند کیفیت را کاهش دهد و دوم، با توجه به اینکه هر لایه مورد عملیات زیرنمونه‌برداری قرار می‌گیرد و اندازه‌ی آن یک چهارم می‌شود بنابراین سربار اعمال ادغام توکن نسبت به کاهش زمانی که ایجاد می‌کند بیشتر است و این موضوع سبب می‌شود که فرآیند ادغام توکن‌ها در ویژگی‌های سطح بالا بهبود عملکرد مطلوب را نداشته باشد.

بعد از تعیین لایه‌ها و بلوک‌های مدنظر در مرحله‌ی بعدی برای اعمال ادغام توکن‌ها از فرآیندی به شکلی که پیش‌تر هم بحث شد استفاده می‌شود. در این فرآیند ابتدا توکن‌ها به یک مجموعه‌ی مبدا و مقصد تقسیم می‌شوند تا از روی آنها گرافی دوبخشی ساخته شود. در اینجا به جای استفاده از رویکرد دوبخشی‌کردن توکن‌ها و ترکیب آنها می‌توانیم از سایر روش‌ها مانند خوشه‌بندی یا برش‌های گرافی استفاده کنیم، اما از آنجاییکه هدف روش ادغام توکن‌ها کاهش محاسبات و زمان آنها است استفاده از روش‌هایی مانند خوشه‌بندی سربار محاسباتی را به شدت افزایش می‌دهد و همچنین ممکن است که در آنها همه‌ی توکن‌ها در تعداد کمی خوشه قرار گیرند که باعث از دست رفتگی حجم زیادی از اطلاعات

[1] ToMe
[2] Hyperparameter

جدول ۳-۱: نتایج اعمال روش ادغام توکن‌ها بر بلوک‌های مبدل لایه‌های مختلف شبکه‌ی یو نتایج این جدول نشان می‌دهند که اعمال ادغام توکن بر لایه‌های میانی باعث کاهش کیفیت می‌شود[۲].

تعداد توکن ورودی	بلوک مبدل	معیار FID↓	زمان(ثانیه)↓
۶۴	۱۵ (همه)	۳۵.۶۶	۱.۵۶
۲۵۶	۱۴	۳۵.۷۱	۱.۵۵
۱۰۲۴	۹	۳۴.۳۷	۱.۵۶
۴۰۹۶	۴	۳۳.۲۹	۱.۶۳

می‌شود. استفاده از گراف دو بخشی این اطمینان را به ما می‌دهد که میزان ادغام از تعداد توکن‌های مقصد کمتر نخواهد شد.

پیش‌تر گفته شد که چهار روش عمده برای تقسیم توکن‌ها به مبدا و مقصد وجود دارد. از بین این روش‌ها انتخاب یک توکن به صورت تصادفی در یک پنجره‌ی ۲ در ۲ بهترین نتیجه‌ی خروجی را می‌دهد و با توجه به اینکه در هر پنجره‌ی ۲ در ۲ یک توکن برداشته می‌شود تعداد توکن‌های مقصد در بهترین حالت یک چهارم تعداد کل توکن‌های تصویر است و در نتیجه‌ی این موضوع در بیشترین کاهش توکن و با فرض اینکه هر توکن مبدا حتما به یک توکن مقصد نگاشت می‌شود ما می‌توانیم ۷۵ درصد کاهش در تعداد توکن‌ها داشته باشیم.

بعد از انتخاب توکن‌های مبدا و مقصد در مرحله‌ی بعدی نیاز داریم که ماتریس مشابهت بین توکن‌های مبدا و مقصد را محاسبه کنیم. معیار مشابهت همانطور که پیش‌تر هم اشاره شد، معیار شباهت کسینوسی است. در مقاله‌ی [۱] در توجیه این انتخاب گفته می‌شود که به واسطه‌ی حضور نویز در ویژگی‌های توکن‌های میانی، استفاده از معیارهایی مانند نرم تفاضل بردار ویژگی‌های توکن‌ها مدل را به اشتباه بیاندازد و در نتیجه معیاری مانند شباهت کسینوسی برای وظیفه‌ی مدنظر ما مناسب‌تر است. در این مرحله ما به ازای هر مبدل یک بار این ماتریس را محاسبه می‌کنیم که محاسبه‌ی خود این ماتریس نیز برای ما سربار محاسباتی خواهد داشت. و همین سربار محاسباتی مخصوصا در لایه‌های میانی و با ویژگی‌های سطح بالا باعث می‌شود که ادغام توکن‌ها برای ما امری به صرفه نباشد.

پس از محاسبه‌ی مشابهت به ازای هر توکن مبدا، مشابه‌ترین توکن مقصد و میزان مشابهت را پیدا می‌کنیم و r درصد از بیشترین مشابهت‌ها را باهم ادغام می‌کنیم. این انتخاب مستقل از میزان مشابهت‌ها، ساختار عکس ورودی و نواحی غیر متغییر تصویر برای همه‌ی گام‌های زمانی و همه‌ی لایه‌ها یکسان است. در ادامه توضیح خواهیم داد که ثابت در نظر گرفتن r یک نقطه‌ی ضعف و عامل عدم انعطاف رویکرد ادغام توکن‌ها است.

در نهایت نیز توکن‌های انتخاب شده برای ادغام با یکدیگر میانگین گرفته می‌شوند که همانطور که در فصل قبل و در قسمت همجوشی توکن‌ها اشاره شد، میانگین‌گیری می‌تواند باعث تغییر در توزیع و نرم ویژگی‌ها شود و کیفیت را کاهش دهد در همین راستا بهتر است که از عملگرهای دیگری استفاده شود.

نکته‌ی دیگری که در توضیح جزئیات مدل پایه باید به آن اشاره کرد، مساله‌ی رفع ادغام[۳] در مدل‌های پخشی است. در این مدل‌ها برخلاف مبدل بینایی که در نهایت یک توکن دسته‌بندی را خروجی می‌دهد در هر گام نویز اضافه‌شده به تصویر پیش بینی می‌شود در همین راستا اندازه‌ی خروجی بلوک مبدل با اندازه‌ی ورودی آن کاملا برابر است. به همین دلیل در روش ادغام توکن در مدل‌های پخشی پس از محاسبه‌ی مبدل باید فرآیند رفع ادغام صورت گیرد. رفع ادغام در روش ادغام توکن‌ها در مدل پخشی به صورت جایگذاری در تک تک توکن‌های ترکیب شده انجام می‌گیرد. به طور دقیق‌تر اگر توکن x_1 و x_2 ادغام شده و در توکن $x_{1,2}$ به صورت زیر قرار گیرند:

$$x_{1,2} = \frac{x_1 + x_2}{2} \tag{۱-۳}$$

پس از محاسبه‌ی مبدل و تبدیل $x_{1,2}$ به $x^*_{1,2}$ برای رفع ادغام داریم:

$$x'_1 = x'_2 = x^*_{1,2} \tag{۲-۳}$$

این روش با وجود اینکه موجب از دست رفتن اطلاعات می‌شود، در شرایطی که توکن‌ها مشابهت زیادی دارند خطای آن کم خواهد بود.
در جمع‌بندی این قسمت باید گفت که روش ادغام توکن‌ها در کنار بهبود خوبی که ایجاد می‌کند و از روش‌های هرس توکن به تنهایی بهتر است، نقاط ضعفی دارد که بهبود آن‌ها می‌تواند این روش را کاراتر سازد. نقاط ضعف اشاره شده را می‌توان در موردهای زیر خلاصه کرد:

- سربار محاسباتی محاسبه‌ی ماتریس مشابهت می‌تواند باعث شود که این روش بخشی از کاهش زمانی را که به واسطه‌ی ادغام توکن‌ها ایجاد می‌کند، استفاده کند. همچنین در لایه‌های عمیق‌تر شبکه‌ی یو و در ویژگی‌های سطح بالاتر اعمال این روش چندان به صرفه نباشد.
- در نظر گرفتن درصد ادغام r به صورت ثابت برای لایه‌ها و بلوک‌ها و گام‌های زمانی مختلف این روش را انعطاف ناپذیر می‌کند. این درصد ثابت همچنین بدون توجه به ویژگی‌های تصویر و فرکانس‌های مشابهت در نظر گرفته می‌شود.

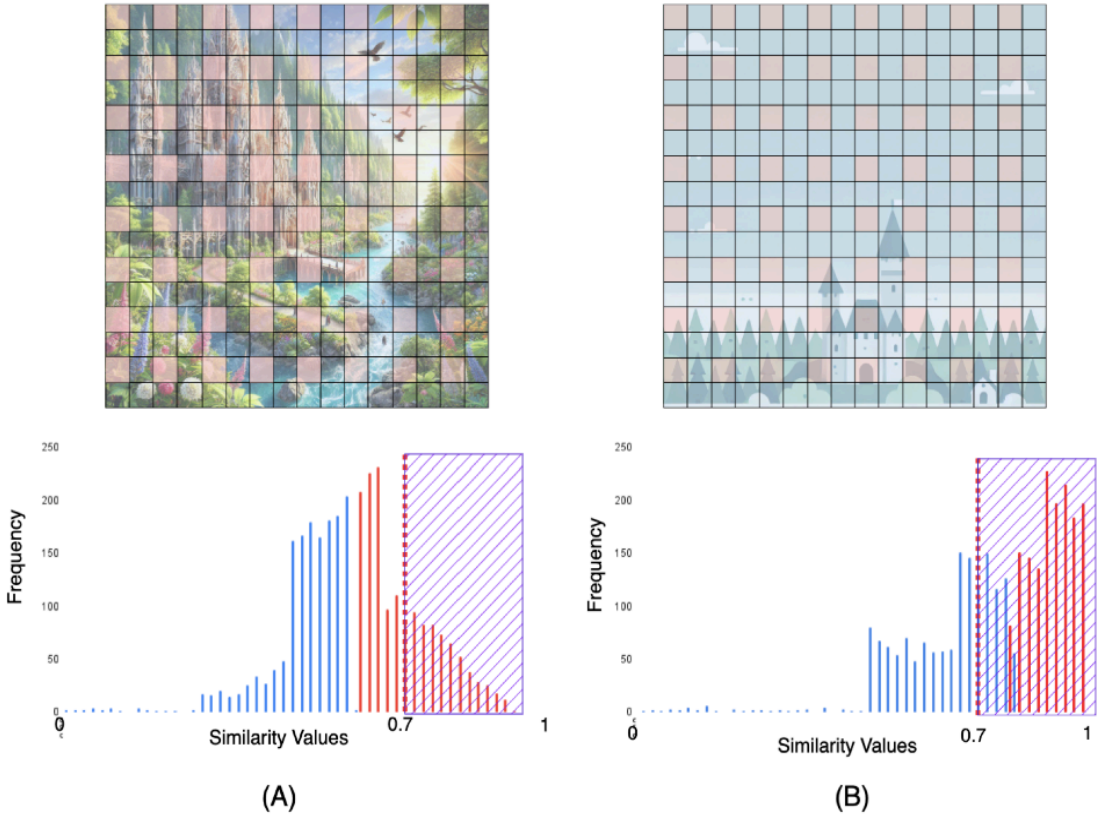
۳-۲ تطبیقی کردن روش ادغام توکن‌ها

در روش پیشنهادی مطرح شده در این پژوهش و در راستای بهبود روش ادغام توکن‌ها اولین ایده تطبیقی[۴] کردن روش پایه است. با توجه به نقاط ضعفی که در قسمت قبل گفته شد، ثابت بودن درصد ادغام در همه‌ی لایه‌ها و همه‌ی گام‌های زمانی و بدون در نظر گرفتن تصویر ورودی باعث عدم انعطاف روش ادغام

[3] Unmerging
[4] Adaptive

توکن‌ها می‌شود.

شکل ۳-۱ این موضوع را به صورت نمادین نشان می‌دهد. در نمودار a هیستوگرام مشابهت توکن‌های مبدا و مقصد یک تصویر در فضای پنهان با توکن‌های متنوع را داریم. این تصویر جزئیات زیادی دارد و تغییرات متعددی در پیکسل‌های متوالی آن وجود دارد. قسمت قرمز نمودار ۵۰ درصد توکن‌ها را که در روش ادغام توکن‌ها با یکدیگر ادغام خواهند شد نشان می‌دهد. همانطور که می‌بینیم در این مثال در روش ادغام توکن‌ها بدون در نظر گرفتن توزیع آنها حجم زیادی از توکن‌هایی که مشابهت خیلی کمتری هم دارند را ادغام می‌کنیم. این امر باعث از دست رفتن اطلاعات در تعدادی از توکن‌های مبدا که به اشتباه با توکن مقصدی که شباهت زیادی با آن ندارند ادغام شده‌اند می‌شود.



شکل ۳-۱: تصویر نمادین تفاوت حالت انطباق‌پذیر و حالت ثابت ادغام توکن‌ها

قسمت A یک تصویر با جزئیات زیاد را نشان می‌دهد در این تصویر مشابهت میان توکن‌ها کم است و بنابراین فرکانس مقادیر مشابهت زیاد کمتر است. قسمت قرمز نمودار ۵۰ درصد توکن‌ها برای ادغام را نشان می‌دهد که همانطور که مشاهده می‌شود در این حالت تعداد زیادی از توکن‌های غیر مشابه نیز با یکدیگر ادغام می‌شوند که این موضوع باعث از دست رفتن اطلاعات و کیفیت تصاویر می‌شود. این در حالی است که با داشتن یک آستانه‌ی ثابت در این حالت فقط توکن‌هایی که واقعا مشابهت بیش از آستانه دارند با یکدیگر ترکیب می‌شوند. قسمت B نیز عکس این سناریو را نشان می‌دهد. تصویر این قسمت یک تصویر با جزئیات کم و در نتیجه تعداد زیادی توکن مشابه می‌باشد، در این حالت روش ادغام توکن‌ها از تمام ظرفیت ادغام استفاده نمی‌کند و تعدادی از توکن‌های مشابه را ادغام نمی‌کند و این در حالی است که روش پیشنهادی توکن‌های با مشابهت بیش از آستانه را ترکیب می‌کند که حجم بیشتری از توکن‌ها در این مثال است.

در حالت دیگر که در قسمت b تصویر ۳-۱ آمده است، نمودار هیستوگرام یک تصویر با جزئیات کم و توکن‌های نسبتا مشابهی را داریم که تغییرات توکن‌ها نیز در این تصویر کم است. در این حالت با توجه به خاصیت تصویر می‌توانیم حجم بیشتری توکن را بدون از دست رفتن اطلاعات ادغام کنیم چراکه توکن‌های مشابه زیادی داریم. در حالیکه ۵۰ درصد توکن‌ها که با قسمت قرمز نمودار مشخص شده‌اند خیلی کمتر از امکان ادغام واقعی هستند. بنابراین در نظر گرفتن یک درصد ثابت برای ادغام بدون اهمیت به توزیع کلی مشابهت‌های موجود می‌تواند باعث عدم کارایی روش ادغام توکن‌ها گردد.

راه‌حل تغییر این عدم انعطاف موجود در روش ادغام توکن‌ها تعریف یک حد آستانه برای انتخاب توکن‌هایی که با هم ادغام می‌شوند می‌باشد. در این راه‌حل با تعریف میزان مناسب از حد آستانه‌ی ادغام می‌توانیم هر تعداد جفت توکنی که از حد آستانه مشابه‌ترند را با یکدیگر ترکیب کنیم. خط‌چین‌های قرمز در قسمت a و b تصویر ۳-۱ آستانه‌ی ۰.۷ را برای ادغام توکن‌ها نشان می‌دهند. در این حالت معایبی که پیش‌تر و به واسطه‌ی تعریف درصد ثابت ادغام با آنها مواجه بودیم برطرف می‌شود. در روش پیشنهادی ما از تعریف یک آستانه‌ی t برای ادغام به جای استفاده از درصد ادغام در روش ادغام توکن‌ها استفاده شده است.

الگوریتم ۳ روش ادغام توکن غیرتطبیقی را نشان می‌دهد. همچنین الگوریتم ۴ نیز روش پیشنهادی را نشان می‌دهد. همانطور که از مقایسه‌ی دو الگوریتم می‌توان دید در این حالت فرآیند محاسبه‌ی گراف دوبخشی در هر دو یکسان است و فقط در مرحله‌ی آخر که فرآیند ادغام صورت می‌گیرد رویکرد متفاوتی پی گرفته می‌شود. همچنین در روش تطبیقی نیازی به مرتب سازی مقادیر یال‌ها نیست که خود می‌تواند نقطه‌ی قوت دیگری برای این روش باشد.

الگوریتم ۳ روش ادغام توکن غیرتطبیقی

۱: درصد ادغام :$= r$ ، تصویر ورودی :$= X$

۲: $src, dst \leftarrow random2x2(X)$ ▷ تقسیم توکن‌های ورودی به دو دسته‌ی مبدا و مقصد

۳: برای هر توکن مبدا $src[i]$ مراحل زیر را محاسبه کن:

۴: $dst[i] \leftarrow argmax_j\,(cosine\ sim(src[i], dst[j]))$ ▷ ذخیره‌ی مشابه‌ترین توکن مقصد به هر توکن مبدا

۵: $edge[i] \leftarrow max_j\,(cosine\ sim(src[i], dst[j]))$ ▷ ذخیره‌ی مقادیر یال‌های گراف دوبخشی

۶: پایان حلقه

۷: یال‌ها را بر اساس مقادیر آنها از کوچک به بزرگ مرتب کن

۸: r یال با بزرگترین مقدار را انتخاب کن و توکن‌های مبدا و مقصد آنها را میانگین بگیر.

الگوریتم ۴ روش ادغام توکن تطبیقی

۱: آستانه‌ی ادغام :$t =$ ، تصویر ورودی :$X =$

۲: $src, dst \leftarrow random2x2(X)$ ▷ تقسیم توکن‌های ورودی به دو دسته‌ی مبدا و مقصد

۳: برای هر توکن مبدا $src[i]$ مراحل زیر را محاسبه کن:

۴: $dst[i] \leftarrow argmax_j\,(cosine\ sim(src[i], dst[j]))$ ▷ ذخیره‌ی مشابه‌ترین توکن مقصد به هر توکن مبدا

۵: $edge[i] \leftarrow max_j\,(cosine\ sim(src[i], dst[j]))$ ▷ ذخیره‌ی مقادیر یال‌های گراف دوبخشی

۶: پایان حلقه

۷: توکن‌های مبدا و مقصد هر یالی را که مقدار آن از آستانه‌ی t بیشتر است با میانگین‌گیری ادغام کن.

۳-۳ استفاده از حافظه‌ی نهان در ادغام توکن‌ها

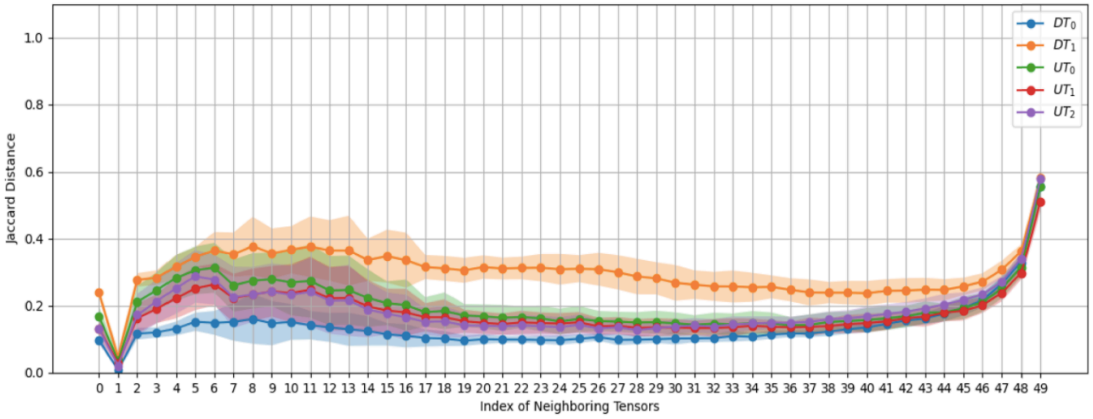
موضوع دیگری که باعث عدم کارایی روش ادغام توکن‌ها می‌شود محاسبه‌ی ماتریس مشابهت توکن‌ها است. این ماتریس با توجه به اینکه برای هر مبدل و در هر گام زمانی محاسبه می‌شود می‌تواند سربار محاسباتی زیادی به روش ادغام توکن‌ها تحمیل کند.
از طرفی با توجه به ایده‌ی اصلی که در عمده‌ی روش‌های مبتنی بر حافظه‌ی پنهان داریم که به افزونگی زمانی و تغییرات بسیار آهسته و هموار در خروجی لایه‌های مختلف اشاره می‌کند می‌توانیم از این موضوع برای کم کردن سربار محاسباتی روش ادغام توکن‌ها استفاده کنیم.
در روش‌های معرفی شده برای رویکرد استفاده از حافظه‌ی پنهان دیدیم که در اکثر قسمت‌های شبکه‌ی یوی نویززدا تغییرات به صورت ناچیز و هموار رخ می‌دهند و اساسا همین مشاهده باعث ایجاد این دسته از روش‌ها است. ایده‌ی اصلی ما نیز در این بخش بر همین مبنا است که اگر میزان تغییرات در خروجی بلوک‌های مختلف شبکه‌ی یو در گام‌های زمانی مختلف بسیار کم است پس الگویی که برای ادغام توکن‌های خروجی هر یک از این بلوک‌ها که در واقع توکن‌های ورودی بلوک بعدی هستند استفاده می‌شود نیز تغییر چندانی نخواهد داشت و بنابراین می‌توانیم به جای محاسبه‌ی ماتریس مشابهت و به دنبال آن جفت‌های توکن‌های مبدا و مقصد، آنها را از حافظه‌ی پنهان به دست آوریم و فقط در برخی از گام‌ها آنها را محاسبه کنیم.
برای بررسی درستی فرضیه‌ی اولیه‌ای که در خصوص ناچیز بودن تغییرات روش ادغام در بلوک‌های متناظر داریم، از معیار فاصله‌ی جاکارد[۵] مجموعه‌ی جفت‌های ادغام شده در دو گام متوالی استفاده می‌کنیم. معیار فاصله‌ی جاکارد برای دو مجموعه‌ی A_n که جفت توکن‌های مشابه در گام n هستند و

[5] Jaccard Distance

A_{n+1} که جفت توکن‌های مشابه در گام $n+1$ هستند، تعریف می‌شود و میزان مشابهت اعضای آنها را به صورت زیر محاسبه می‌کند:

$$JaccardDistance(A_n, A_{n+1}) = 1 - \frac{|A_n \cap A_{n+1}|}{|A_n \cup A_{n+1}|} \tag{۳-۳}$$

این معیار برای گام‌های زمانی صد عکس از صد کلاس مختلف $ImageNet$ و در پنج بلوک مبدل با $MaxDownsample = 1$ در شکل ۳-۲ رسم شده است. همانطور که مشاهده می‌کنید میزان تغییرات در اکثر گام‌ها چندان چشم‌گیر نیست و این موضوع فرضیه‌ی ما مبنی بر هموار بودن تغییرات ماتریس مشابهت و جفت توکن‌های ادغامی را تایید می‌کند.



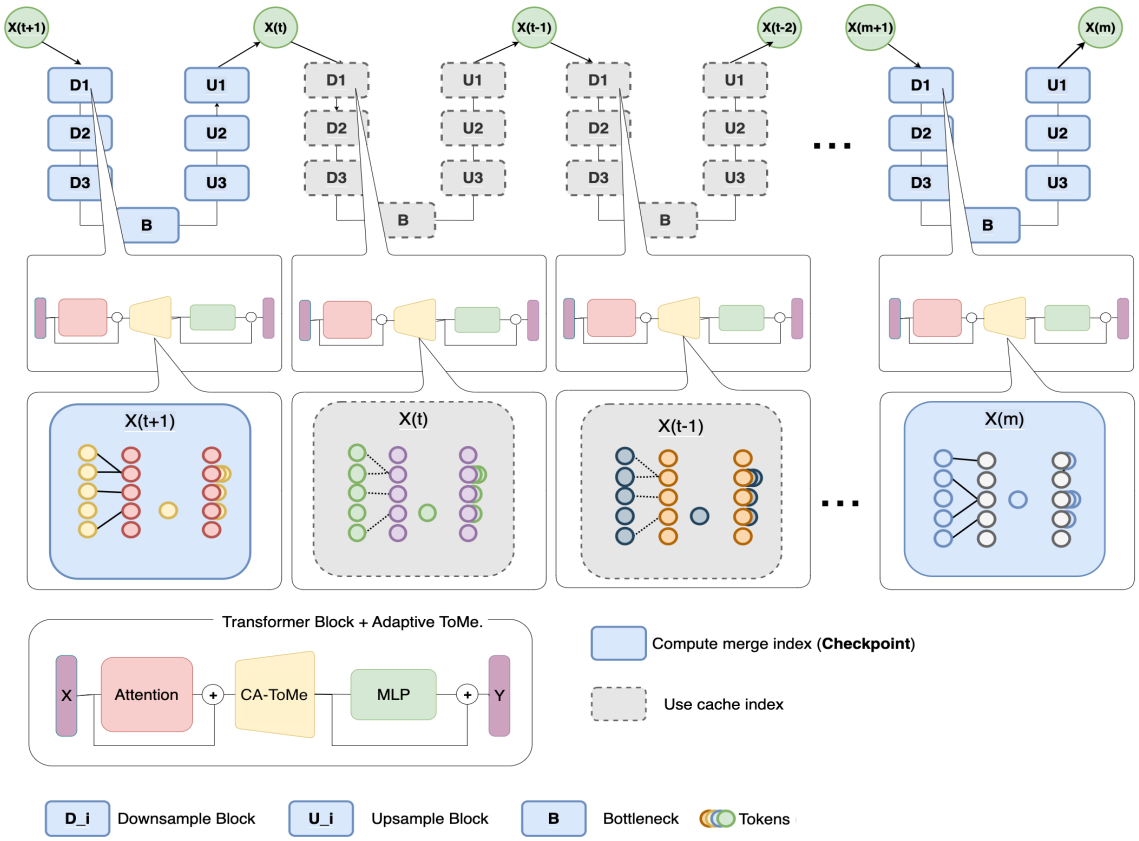
شکل ۳-۲: نمودار فاصله‌ی جاکارد مجموعه‌ی جفت‌های ادغام در روش ادغام توکن‌ها

این نمودار فاصله‌ی جاکارد جفت‌های ادغام هر دو گام متوالی برای صد عکس تولید شده از صد کلاس $ImageNet$ را نشان می‌دهد. همچنین نواحی هاشور خورده اطراف هر نمودار واریانس مقادیر را نشان‌می‌دهد.

بر اساس این مشاهده می‌توانیم فرآیند کامل ادغام را در تعدادی از گام‌ها انجام دهیم و در سایر گام‌ها از نمایه‌های جفت‌های ادغام شده در مراحل قبلی استفاده کنیم و بدون محاسبه‌ی مجدد ماتریس مشابهت آنها را با یکدیگر ادغام کنیم. شکل ۳-۳ این روش را به صورت شماتیک نشان می‌دهد. در این شکل گام‌هایی که با رنگ آبی مشخص شده‌اند بیانگر گام‌هایی هستند که محاسبات در آنها به صورت کامل انجام می‌شود و گام‌هایی که با رنگ خاکستری مشخص شده‌اند گام‌هایی را مشخص می‌کنند که در آنها جفت‌های قابل ادغام از حافظه‌ی پنهان خوانده می‌شود.

در نهایت، روش پیشنهادی که روش ادغام توکن تطبیقی مبتنی بر حافظه‌ی پنهان نام دارد، در راستای برطرف کردن نقاط ضعف روش ادغام توکن ارائه شده است. این روش به طور کلی دو نوآوری اصلی دارد:

- تطبیقی کردن فرآیند ادغام با تعریف یک آستانه برای میزان مشابهت توکن‌هایی که برای ادغام برگزیده می‌شوند.



شکل ۳-۳: شمای کلی روش پیشنهادی و استفاده از حافظه‌ی پنهان در فرآیند ادغام توکن‌ها

در این شکل بلوک روش پیشنهادی ما با $CA-ToMe$ نشان داده‌شده است که مخفف همان روش پیشنهادی با نام ادغام توکن تطبیقی مبتنی بر حافظه‌ی پنهان است. همچنین گام‌های آبی نمایانگر گام‌هایی هستند که در آنها محاسبات انجام می‌شود و گام‌های خاکستری نمایانگر گام‌هایی هستند که در آنها محاسبه از حافظه‌ی پنهان خوانده می‌شود.

- استفاده از حافظه‌ی پنهان بر مبنای این ایده که تغییرات توکن‌ها در گام‌های متوالی ناچیز است و ذخیره‌ی ماتریس مشابهت و نمایه‌های جفت‌های قابل ادغام در حافظه‌ی پنهان.

فصل چهارم

آزمایش‌ها و نتایج

در این فصل ابتدا تنظیمات آزمایش‌ها و روش‌های ارزیابی را معرفی می‌کنیم و سپس به ارائه‌ی نتایج و تحلیل و بررسی آنها می‌پردازیم.

۴-۱ مجموعه داده

به طور کلی و در مقالات مطالعه شده، هر نوع داده‌ی تصویری حجیم دارای برچسب برای ارزیابی امر تولید تصویر به کار گرفته می‌شود اما دو دسته داده در اکثر پژوهش‌های اخیر در بحث استفاده از حافظه‌ی پنهان و کاهش توکن به کار گرفته شده‌اند که در ادامه هر یک را معرفی می‌کنیم و فرآیند ارزیابی تولید تصویر با هر یک را ذکر می‌کنیم.

۴-۱-۱ مجموعه داده‌ی $ImageNet$

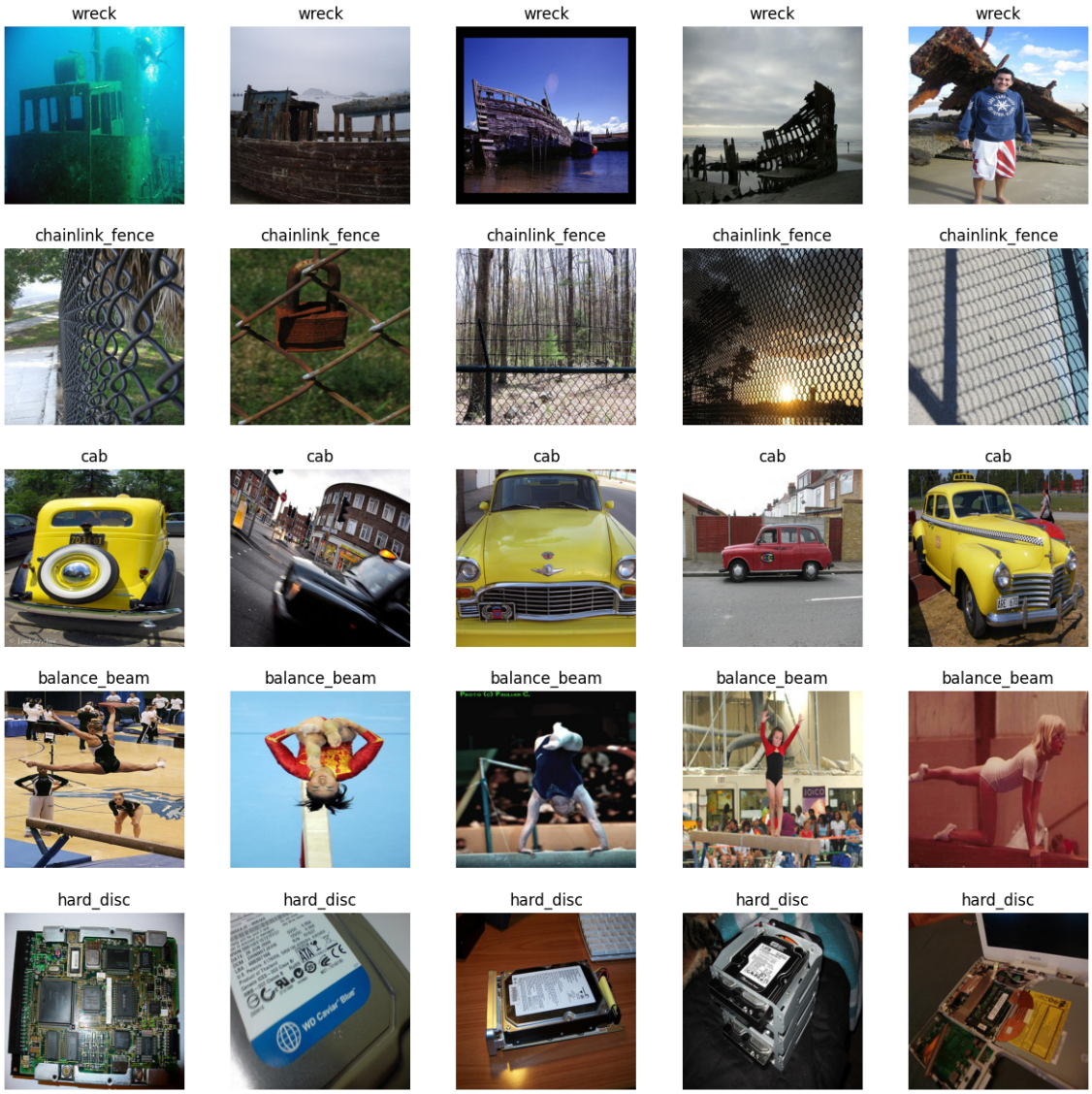
مجموعه داده‌ی $ImageNet$ [۴] یک مجموعه داده‌ی تصویری در مقیاس بزرگ و به صورت عمومی در دسترس است که بر اساس ساختار شبکه‌ی کلمات[۱] توسعه داده شده است. در شبکه‌ی کلمات هر مفهوم معنادار با تعدادی کلمه‌ی هم‌معنی تعریف می‌شود که به آن مجموعه‌ی هم‌معنی[۲] گفته می‌شود و در حدود ۸۰ هزار مجموعه‌ی هم‌معنی موجود در شبکه‌ی کلمات اسم هستند. تلاش سازندگان مجموعه داده‌ی $ImageNet$ بر این است که به ازای هر مجموعه‌ی هم‌معنی موجود حدود ۱۰۰۰ تصویر جمع‌آوری کنند.

پرکاربردترین زیرمجموعه از $ImageNet$ که در کاربردهای مختلفی از دسته‌بندی گرفته تا تولید تصاویر به کار می‌رود، مجموعه‌ای است شامل ۱۰۰۰ کلاس داده‌ی مختلف و شامل یک میلیون و دویست هزار داده‌ی آموزشی ۵۰ هزار داده‌ی صحت سنجی و ۱۰۰ هزار داده‌ی آزمون. این مجموعه داده در سایت $Kaggle$ در دسترس است و مجموعه داده‌ای است که در ارزیابی ما نیز مد نظر بوده است. شکل ۴-۱ تعدادی از تصاویر این مجموعه داده و نام کلاس آنها را نشان می‌دهد.

۴-۱-۲ مجموعه داده‌ی $COCO$

مجموعه داده‌ی $COCO$[۱۶] یا اشیای رایج در پس زمینه‌ی ماکروسافت[۳] یک مجموعه داده‌ی در مقیاس بزرگ دیگر است که برای انواعی از مسائل بینایی ماشین از قطعه‌بندی معنایی، شناسایی اشیا تا مسئله‌ی تولید تصاویر به کار گرفته می‌شود. این مجموعه شامل برچسب اشیا، مستطیل دربرگیرنده‌ی آنها و متن[۴] توصیف کننده‌ی اشیا است و در مجموع شامل بیش از ۳۳۰ هزار تصویر و بیش از ۸۰ شی مختلف

[1] WordNet
[2] Synonym Set
[3] Microsoft Common Objects in Context
[4] Caption

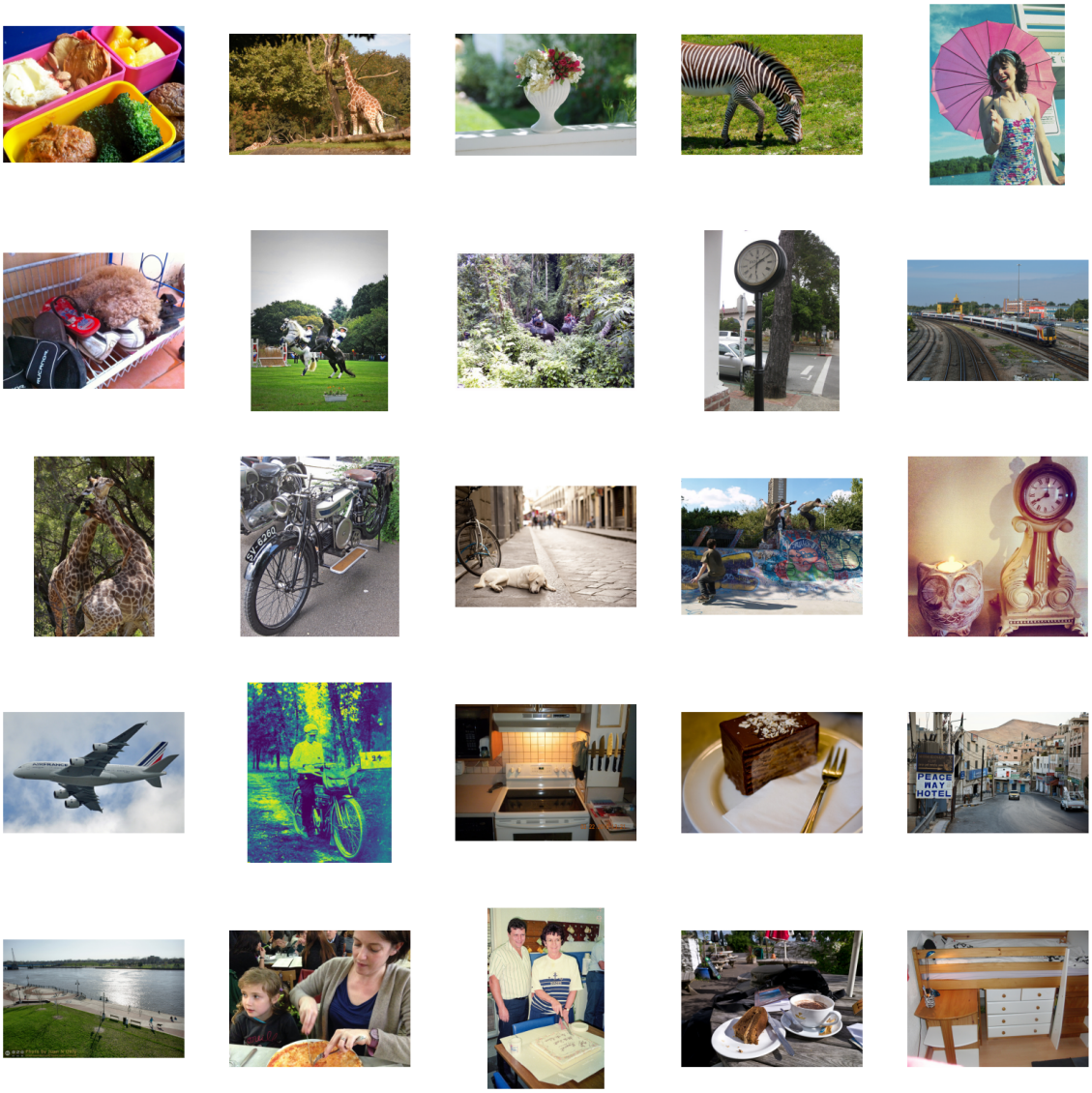


شکل ۴-۱: نمونه تصاویری از مجموعه داده‌ی $ImageNet$

این تصویر مجموعه‌ی ۵ تصویر از ۵ دسته‌ی مجموعه داده‌ی $ImageNet$ را نشان می‌دهد[۴].

می‌باشد. تصویر ۴-۲ تعدادی از تصاویر این مجموعه داده را نشان می‌دهد.

علاوه بر این دو مجموعه‌داده، مجموعه داده‌های دیگری مانند $CIFAR10$ [۱۲]، $LSUN-bedroom$ و $LSUN-Church$ نیز وجود دارند که به علت کم بودن تعداد اشیا و کلاس‌های موجود در آنها و خاص منظوره بودن آنها در این ارزیابی لحاظ نشده‌اند.



شکل ۴-۲: نمونه تصاویری از مجموعه داده‌ی $COCO$
این تصویر مجموعه‌ی ۲۵ تصویر از مجموعه داده‌ی $coco$ است[۱۶].

۴-۲ معیارهای ارزیابی

مساله‌ی مطرح در این پژوهش دو معیار اساسی و مهم دارد که این دو معیار در یک مصالحه هستند. به طور کلی هدف ما افزایش سرعت است و همانطور که در قسمت‌های پیشین گفته شد این افزایش سرعت از طریق کاهش حجم محاسبات شبکه‌ی نویززدا به دست می‌آید. اما نکته‌ی مهم این است که این کاهش حجم محاسبات نباید کیفیت تصاویر تولیدی را تحت‌الشعاع قرار دهد. از این رو دو معیار اساسی در این پژوهش زمان و کیفیت تصاویر تولیدی هستند.

برای اندازه گیری زمان همان زمان میانگین تولید تصاویر در فرآیند نمونه برداری در نظر گرفته شده‌است. عیب اصلی این معیار این است که بر حسب نوع سخت‌افزار مورد استفاده ممکن است تفاوت کند. در

این آزمایش تمامی نتایج بر روی پردازشگر گرافیکی $Tesla\ V100\ GPU$ انجام شده است و زمان‌های گزارش شده اعدادی هستند که در این تنظیمات سخت افزاری مشاهده شده‌اند. همچنین میزان کاهش زمان نسبت به مدل پایه نیز به صورت ضریب گزارش شده است تا حالت نرمال شده را نیز داشته باشیم.
برای تعیین کیفیت از معیار فاصله‌ی اکتسابی فرچت[۵] یا FID [۷] استفاده می‌کنیم. این معیار از مشابهت آماری بین تصاویر اصلی و تصاویر تولید شده برای ارزیابی کیفیت تصاویر تولید شده استفاده می‌کند. این معیار با به دست آوردن بازنمایی ویژگی‌های تصاویر دو مجموعه با استفاده از یک شبکه‌ی از پیش آموزش دیده‌ی $inception$ و محاسبه‌ی فاصله‌ی فرچت روی این بازنمایی‌های به دست آمده، معیار مد نظر را محاسبه می‌کند. فاصله‌ی فرچت، بر اساس میانگین و کوواریانس مقایسه می‌کند که توزیع ویژگی‌های تصاویر تولید شده تا چه اندازه به تصاویر اصلی نزدیک است. رابطه‌ی ریاضیاتی آن به صورت زیر می‌باشد:

$$FID = \|\mu_r - \mu_g\|^2 + Tr(\Sigma_r + \Sigma_g - 2(\Sigma_r\Sigma_g)^{1/2}), \qquad (۱-۴)$$

که در این فرمول μ_r و μ_g به ترتیب میانگین بازنمایی ویژگی‌های تصاویر اصلی و تصاویر تولید شده هستند. همچنین Σ_r و Σ_g هم ماتریس کوواریانس این ویژگی‌ها هستند. هر چقدر مقدار FID کمتر باشد نمایان‌گر کیفیت بهتر تصاویر تولیدی و شباهت بیشتر آن با مجموعه‌ی تصاویر واقعی می‌باشد. این معیار همبستگی خوبی با درک بینایی انسان از کیفیت دارد و به عنوان معیار اصلی برای مقایسه‌ی کیفیت تصاویر در مجموعه داده‌های مختلف مورد استفاده است.
با وجود این که در اکثر مقالات مرتبط با تولید تصاویر معیار FID گزارش می‌شود؛ این معیار محدودیت‌های زیادی دارد که به برخی از آنها در ادامه اشاره می‌کنیم:

- این معیار یک معیار آماری است و با توجه به اینکه از تخمین میانگین و کوواریانس استفاده می‌کند، نیاز داریم که حجم داده‌ی ورودی و نمونه‌های زیادی داشته باشیم تا محاسبات قابل اتکا باشد.

- با وجود اینکه معیار FID از نظر کلیات مشابهت دو مجموعه داده با درک بینایی انسان همبستگی دارد در جزئیات و مصنوعات اضافه شده به تصویر می‌تواند تفاوت زیادی داشته باشد.

- شرایط اندازه‌گیری و پیش‌پردازش‌های انجام شده در محاسبه‌ی این معیار، در مقدار آن موثر است به این معنا که به ازای اندازه‌ی نمونه‌های مختلف و پیش‌پردازش‌های متفاوتی از تصاویر خروجی از یک شبکه می‌توانیم معیار FID متفاوتی بگیریم.

برای رفع محدودیت‌های معیار FID دو رویکرد به کار گرفته می‌شود. اول، ثابت نگه داشتن شرایط آزمایش برای مدل‌های مبتنی بر یک مدل پایه و دوم، استفاده از سایر معیارها مانند $LPIPS$ [۳۵] که در ادامه به کارگیری هر دوی این رویکردها در بهبود ارزیابی آزمایشات این پژوهش را توضیح می‌دهیم.

[5] Frechet Inception Distance

برای یکسان‌سازی شرایط آزمایش با سایر آزمایش‌های افزایش سرعت مدل‌های پخشی بسته به نوع داده دو روش وجود دارد. در محاسبه‌ی FID بر روی داده‌ی $COCO$ به روش استاندارد، نیاز به تولید ۳۰ هزار عکس و مقایسه‌ی آنها با مجموعه‌ی صحت‌سنجی مجموعه داده‌ی $COCO$ داریم. برای محاسبه‌ی FID در مجموعه داده‌ی $ImageNet$ یک روش استاندارد وجود دارد که در مقالات کاهش توکن از این روش استفاده می‌شود. در این روش از هر کلاس مجموعه داده‌ی $ImageNet$ دو تصویر با فرمان "$A\ High\ Quality\ Image\ of\ (ClassLabel)$" تولید می‌کنیم که در مجموع ۲۰۰۰ عکس می‌شود و سپس برای مجموعه داده‌ی اصلی نیز ۵ تصویر از هر کلاس را از مجموعه‌داده‌ی آموزش می‌خوانیم که در مجموع ۵۰۰۰ عکس می‌شود. سپس فاصله‌ی اکتسابی فرچت را برای این دو مجموعه داده محاسبه می‌کنیم. در این شرایط که آزمایش ما هم به همین صورت انجام شده است، نتایج قابلیت مقایسه پذیری با روش‌های کاهش توکن دارند.

برای افزایش دقت محاسبات یک معیار دیگر را نیز در تصمیم گیری در خصوص انتخاب بهترین تنظیمات و نیز در بحث مقایسه‌ی مدل‌ها در نظر گرفتیم. این معیار مشابهت ادراکی یادگرفته شده‌ی قطعات تصویر[۶] یا $LPIPS$ [۳۵]نام دارد و برای سنجش کیفیت با معیاری نزدیکتر به درک بینایی انسان به کار می‌رود.

معیار $LPIPS$ شباهت ادراکی میان دو تصویر را با استفاده از ویژگی‌های شبکه‌های عصبی عمیق محاسبه می‌کند. این معیار با عبور تصاویر از شبکه‌های پیچشی عمیق از پیش آموزش دیده[۷] مانند $AlexNet$[۱۳] یا VGG [۲۸] بازنمایی ویژگی‌های آنها را از لایه‌های مشخصی استخراج می‌کند. مفهوم قطعات تصویر در نام‌گذاری این معیار نیز به همین بازنمایی‌های استخراج شده از لایه‌های مختلف اشاره دارد. سپس این بازنمایی‌ها نرمال‌سازی می‌شوند و در نهایت معیار فاصله‌ی نرم دوی وزن‌دار روی این بازنمایی‌های نرمال‌شده محاسبه می‌شود. فرمول این فاصله به صورت زیر است:

$$d(x, x_0) = \sum_l \frac{1}{H_l W_l} \sum_{h,w} \|w_l \odot (y^l_{hw} - y^l_{0hw})\|_2^2 \qquad (۲-۴)$$

که در این فرمول فاصله‌ی بین دو تصویر x_0 و x با استفاده از شبکه‌ی $\mathcal{F}$ محاسبه می‌شود و در این شبکه ما ویژگی‌ها را از L لایه استخراج می‌کنیم و در راستای بُعد کانال نرمال‌سازی می‌کنیم که به ما خروجی‌های $y^l, y^l_0 \in \mathbb{R}^{H_l \times W_l \times C_l}$ برای لایه‌ی l را می‌دهد. همچنین برای وزن‌های نرم دو داریم $w_l = 1 \forall l$ که معادل محاسبه‌ی فاصله‌ی کسینوسی است. با توجه به اینکه این معیار فاصله را محاسبه می‌کند، هر چقدر که کمتر باشد نمایانگر کیفیت تصاویر تولیدی می‌باشد. این معیار در کنار معیار FID می‌تواند برای توجیه نتایج و بررسی کیفیت در جزئیات و مصنوعات تصویر به کار گرفته شود.

[6] Learned Perceptual Image Patch Similarity

[7] Pre-trained Deep Convolutional Networks

۴-۳ تعیین ابرپارامترها

در این بخش برای تعیین ابرپارامترهای روش پیشنهادی از مدل $Stable\ Diffusion$ نسخه‌ی ۱.۵ استفاده شده است اما نتایج نهایی و پس از انتخاب بهینه‌ی ابرپارامترها روی مدل $Stable\ Diffusion$ نسخه‌ی ۲ هم اجرا شده است. بنابراین ابتدا به تفاوت‌های پیاده‌سازی این دو مدل و علت بررسی مدل دوم هم می‌پرداریم.

مدل‌های نسخه‌ی ۱.۵ و ۲ در کلیات معماری مشابه هستند، تفاوت‌های بنیادین این دو مدل در استفاده از کدگذارهای متنی متفاوت و مجموعه‌های آموزشی متفاوت است. در نسخه‌ی ۱.۵ از تعبیه‌های متنی مدل $OpenAI'sCLIP\ ViT-L/14$ [۲۲] استفاده شده است در حالیکه در نسخه‌ی ۲ از $OpenCLIP$ [۳] استفاده شده است که هم متن‌باز[۸] است و هم تقریبا ۵ برابر بزرگتر از مدل قبلی است و توانایی فهم بهتر و سازگاری فرمان و تصویر بهتری را دارد. علاوه بر این مدل نسخه‌ی ۲ بر روی زیرمجموعه‌ای از مجموعه داده‌ی $LAION-5B$ آموزش داده شده است که شامل تصاویر با معیار هنری کم و تصاویر نامناسب نیست و از این رو در تولید تصاویر هنری نسبت به نسخه‌ی ۱.۵ ضعیف تر عمل می‌کند. همچنین ابعاد پیش‌فرض تولید تصویر در نسخه‌ی ۲ به ۷۶۸ در ۷۶۸ تغییر یافته است که در نسخه‌ی ۱.۵ این عدد ۵۱۲ در ۵۱۲ است. علت استفاده از این مدل برای ارزیابی نتایج نهایی این است که نسخه‌ی ۱.۵ از کتابخانه‌ی $Huggingface$ حذف شده است و امکان استفاده از وزن‌های آن در زمان نگارش این گزارش وجود ندارد از همین‌رو نتایج بر روی مدل دیگری نیز محاسبه شده‌اند تا کارایی روش پیشنهادی بر روی مدل‌های موجود قابل ارزیابی باشد.

برای بررسی نتایج تعیین ابرپارامترها از مدل $Stable\ Diffusion$ نسخه‌ی ۱.۵ استفاده شده است و تغییرات بر روی تابع رو به جلوی[۹] شبکه‌ی یوی این مدل اعمال شده است. همانطور که در بخش روش پیشنهادی توضیح داده شد، روش ما نیاز به تعیین دو ابرپارامتر دارد. اول مقدار آستانه‌ی t برای تطبیقی کردن فرآیند ادغام توکن‌ها و نیز پارامتر cal_steps که تعیین کننده‌ی نمایه‌ی گام‌هاییست که در آنها فرآیند ادغام توکن‌ها مجدد محاسبه می‌شود.

برای تعیین آستانه‌ی t با توجه به اینکه مقدار مشابهت کسینوسی بین صفر و یک است و با توجه به این موضوع که به ازای هر توکن مبدا توکن مقصدی که بیشترین مشابهت را با آن دارد انتخاب می‌شود و فاصله‌ی این دو در نظر گرفته می‌شود، آستانه معمولا از مقدار ۰/۴ بیشتر و از یک کمتر است. بنابراین در این پژوهش با استفاده از روش تجربی و محاسبه‌ی معیارهای کیفیت و زمان برای آستانه‌های مختلف مقدار این ابرپارامتر تعیین شده است.

جدول ۴-۱ نتایج آزمایشات برای حالتی که حافظه‌ی پنهان در آن استفاده نشده است و آستانه‌های مختلف را نشان می‌دهد. انتظار داریم که با افزایش آستانه که به این معنا است که اجازه‌ی ادغام را فقط برای توکن‌های بسیار مشابه می‌دهیم و شرط سخت‌گیرانه‌تری برای ادغام داریم، زمان هم افزایش پیدا کند چراکه توکن‌های کمتری ادغام می‌شوند و ورودی مبدل کاهش کمتری دارد. از طرفی با توجه به

[8] Open-source

[9] Forward Function

اینکه ادغام بیش از حد باعث از دست رفتن اطلاعات می‌شود، انتظار داریم که با افزایش آستانه که به معنی کاهش تعداد توکن‌های ادغام شده است، معیار FID و $LPIPS$ کاهش پیدا کند که یعنی کیفیت تصاویر تولیدی بهتر شده است. در مقابل در آستانه‌های پایین‌تر اجازه‌ی ادغام تعداد بیشتری از توکن‌ها، حتی با شباهت کمتر را می‌دهیم که باعث کاهش زمان و کیفیت می‌شود. بنابراین باید آستانه‌ای انتخاب شود که مصالحه‌ای میان کیفیت و زمان باشد.

جدول ۴-۱: معیارهای کیفیت و زمان برای آستانه‌های مختلف در روش ادغام تطبیقی در مدل $Stable\ Diffusion\ v1.5$

روش	آستانه	FID	میانگین زمان(ثانیه)	$LPIPS$
ادغام توکن تطبیقی	۰/۴	۳۵/۲۸	۶/۰۷ ± ۰/۰۰۷	۰/۸۵۴۰
	۰/۵	۳۵/۴۶	۶/۰۷ ± ۰/۰۰۴	۰/۸۵۳۴
	۰/۶	۳۵/۵۶	۶/۱۰ ± ۰/۰۰۵	۰/۸۵۲۰
	۰/۷	۳۴/۳۰	۶/۲۳ ± ۰/۰۰۲	۰/۸۴۴۰
	۰/۸	۳۳/۸۰	۶/۵۸ ± ۰/۰۰۴	۰/۸۳۸
	۰/۹	۳۳/۴۲	۶/۹۲ ± ۰/۰۰۳	۰/۸۲۲
	۱/۰	۳۳/۶۶	۷/۵۱ ± ۰/۰۰۴	۰/۸۱۶
ادغام توکن غیرتطبیقی	-	۳۴/۱۶	۶/۳۹ ± ۰/۰۰۶	۰/۸۵۱

همانطور که در جدول ۴-۱ مشاهده می‌شود، تغییرات معیار زمان با آنچه که انتظار می‌رفت مطابقت می‌کند از این جهت که با افزایش آستانه و اعمال سخت‌گیری بیشتر بر ادغام توکن‌های مشابه، کاهش تعداد توکن‌ها کمتر شده و کاهش زمان نیز به دنبال آن کمتر می‌شود. در آستانه‌ی ۱/۰ تقریبا همان مدل پایه‌ی $SDv1.5$ را داریم با این تفاوت که اگر توکن‌هایی عینا مشابه یکدیگر باشند با یکدیگر ترکیب می‌شوند و این در بعضی از تصاویر دارای پس زمینه‌ی ثابت می‌تواند موثر باشد. از آستانه‌ی ۰/۸ تطبیقی کردن روش ادغام تاثیری در کاهش زمان ندارد و نسبت به حالت غیر تطبیقی عملکرد بدتری دارد چرا که آستانه به شدت سخت‌گیرانه است و اجازه‌ی ادغام تعداد زیادی از توکن‌ها را نمی‌دهد.
در بررسی کیفیت در روش تطبیقی، هر دو معیار $LPIPS$ و FID مطابق انتظار تغییر می‌کنند. با افزایش آستانه از ۰/۴ تا ۱/۰ انتظار داریم که کیفیت تصاویر تولیدی افزایش پیدا کند چراکه ادغام توکن‌ها باعث از دست رفتن اطلاعات می‌شود و هر چقدر توکن‌هایی که مشابهت قابل توجهی با یکدیگر ندارند با هم ترکیب شوند کیفیت افت بیشتری خواهد داشت. در بررسی معیار FID به تنهایی از آستانه‌ی ۰/۸ به بعد تطبیقی کردن نسبت به رویکرد غیر تطبیقی برتری دارد اما با اضافه کردن معیار $LPIPS$ مشاهده می‌کنیم که کیفیت تصاویر تولیدی در آستانه‌ی ۰/۷ نیز نسبت به حالت غیرتطبیقی بهتر است.
در مجموع و با در نظر گرفتن مصالحه‌ی زمان و کیفیت می‌توان نتیجه گرفت که آستانه‌ی ۰/۷ از لحاظ زمانی و کیفی نسبت به حالت غیر تطبیقی برتری دارد. در نتیجه در ادامه‌ی آزمایشات این عدد به عنوان بهترین حد آستانه برای تطبیقی ساختن روش ادغام توکن‌ها استفاده می‌شود.

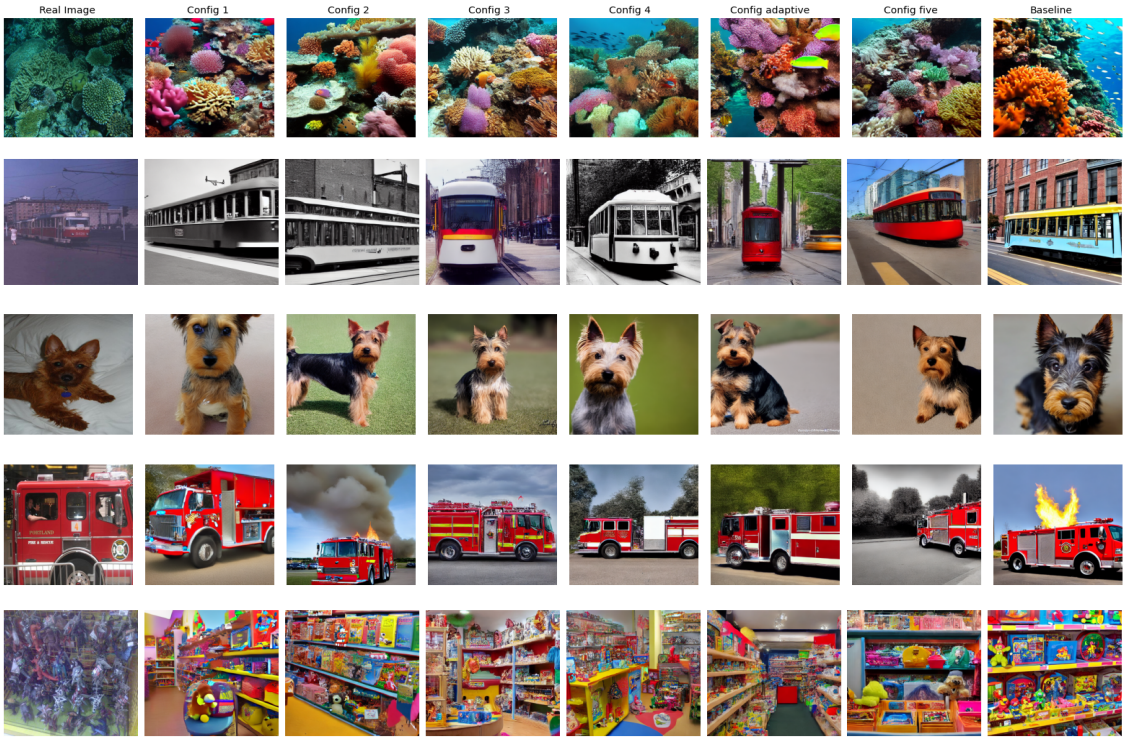
در تعیین گام‌های محاسبه‌ی ادغام توکن‌ها از سه منبع برای تعیین تنظیمات استفاده شده است. در ساده‌ترین رویکرد و مشابه آنچه در رابطه‌ی ۲-۱۷ آمده است، از تعدادی گام با فاصله‌ی مساوی برای محاسبه‌ی ادغام توکن‌ها استفاده شده است. در تنظیمات دوم با توجه به نمودار فاصله‌ی جاکارد تعدادی حالت در نظر گرفته شده است. نمودار ۳-۲ فاصله‌ی جاکارد مجموعه‌ی جفت‌های ادغام در گام‌های متوالی را نشان می‌دهد. در این نمودار ابتدا فاصله زیاد است به این معنا که جفت‌های ادغام در گام‌های اولیه تغییرات زیادی دارند و سپس این تغییرات در گام‌های میانی کاهش و در گام‌های نهایی مجددا افزایش پیدا می‌کند. و در نهایت با مرور کارهای پیشین [۱۸][۱۴] برخی از تنظیمات این روش‌ها نیز به تنظیمات مورد ارزیابی اضافه شده‌اند. این تنظیمات در جدول ۴-۲ آمده است. به عنوان مثال $CONFIG_1$ با فرض اینکه تغییرات در گام‌های ابتدایی بیشتر است و به مرور زمان جزئی‌تر می‌شود، در گام‌های ابتدایی محاسبات بیشتری لحاظ می‌کند و در گام‌های انتهایی تماما از حافظه‌ی پنهان استفاده می‌کند.

جدول ۴-۲: گام‌های محاسبه در تنظیمات مختلف در نظر گرفته شده در روش ادغام توکن تطبیقی مبتنی بر حافظه‌ی پنهان

نام تنظیمات	گام‌های محاسبه
$CONFIG_1$	$[0, 1, 2, 3, 5, 10, 15, 25, 35]$
$CONFIG_2$	$[0, 10, 11, 12, 15, 20, 25, 30, 35, 45]$
$CONFIG_3$	$[0, 8, 11, 13, 20, 25, 30, 35, 45, 46, 47, 48, 49]$
$CONFIG_4$	$[0, 9, 13, 14, 15, 28, 29, 32, 36, 45]$
$CONF_{ADAPTIVE}$	$[0, 1, 5, 7, 10, 12, 15, 35, 40, 45, 46 - 51]$
$CONFIG_{Five}$	$[0, 5, 10, 15, 20, 25, 30, 35, 40, 45, 50]$

برای هر یک از تنظیمات در نظر گرفته شده و با تعیین مقدار آستانه با عدد۰/۷ که به عنوان آستانه‌ی بهینه انتخاب شده است، آزمایشات تولید ۲۰۰۰ تصویر از کلاس‌های مجموعه داده‌ی $ImageNet$ انجام شده است و نتایج این آزمایشات در جدول ۴-۳ آمده‌است. همانطور که مشاهده می‌شود، در میان تنظیمات مختلف حالت $Config_3$ بهترین خروجی را از نظر زمانی و کیفیت تصاویر تولیدی دارد. این معیار از نظر کیفیت و مقدار عددی FID و $LPIPS$ نسبت به سایر روش‌های کمترین مقدار و بیشترین نزدیکی به مدل پایه را دارد همچنین از نظر زمانی نیز تقریبا معادل حالت ۰/۴ در حالت آستانه‌گذاری به تنهایی است که نشان می‌دهد در صورتی که ما از حداکثر امکان ادغام استفاده کنیم، به عدد زمانی نزدیک به این عدد خواهیم رسید.

شکل ۴-۳ تعدادی از این تصاویر تولیدی با تنظیمات مختلف و حالت پایه را نشان می‌دهد. در این تصاویر می‌توانیم از مقایسه‌ی تنظیم $config_3$ با سایر تنظیمات به وجود جزئیات و نبود مصنوعات و بعضا تضاد و تنوع رنگی بهتر اشاره کنیم.



شکل ۴-۳: نمونه تصاویر تولیدی برای تنظیمات مختلف جدول ۴-۲ و به ازای ۵ کلاس مختلف از $ImageNet$

جدول ۴-۳: عملکرد کلی روش ادغام توکن مبتنی بر حافظه‌ی پنهان در تنظیمات گام‌های محاسبه‌ی متفاوت

روش	آستانه/درصد	گام‌های حافظه‌ی پنهان	میانگین زمان(ثانیه)	FID	$LPIPS$
روش پیشنهادی	۰/۷	$CONF_1$	۶/۱۸ ± ۰/۰۲	۳۵/۱۴	۰/۸۴۲۳
روش پیشنهادی	۰/۷	$CONF_2$	۶.۱۳± ۰.۰۰۱	۳۳/۸۴	۰/۸۳۶۵
روش پیشنهادی	۰/۷	$CONF_3$	۶/۰۹ ± ۰/۰۰۱	۳۳/۴۹	۰/۸۳۴۷
روش پیشنهادی	۰/۷	$CONF_4$	۶/۱۲ ± ۰/۰۰۱	۳۴/۱۸	۰/۸۳۷۸
روش پیشنهادی	۰/۷	$CONF_{ADAPTIVE}$	۶/۱۹ ± ۰/۰۰۱	۳۴/۷۷	۰/۸۴۸۵
روش پیشنهادی	۰/۷	$CONF_{FIVE}$	۶/۱۴ ± ۰/۰۰۱	۳۴/۵۷	۰/۸۴۲۹
ادغام توکن‌ها	۰/۵	بدون حافظه پنهان	۶/۳۹ ± ۰/۰۰۶	۳۴/۱۶	۰/۸۵۱۰
مدل پخشی پایه	-	بدون حافظه پنهان	۷.۶۱ ± ۰/۰۰۱	۳۳.۳۷	۰/۸۱۳۶

۴-۴ تحلیل و بررسی نتایج

نتایج نهایی جمع‌بندی شده با بهترین تنظیمات هر مدل برای مدل $Stable\ Diffusion\ v1.5$ در جدول ۴-۴ و برای مدل $Stable\ Diffusion\ v2$ در جدول ۴-۵ آمده است. جدول ۴-۴ نشان می‌دهد که روش ادغام توکن مبتنی بر حافظه‌ی پنهان با حفظ کیفیت می‌تواند در مدل $Stable\ Diffusion\ v1.5$ با ضریب ۱/۲۴ افزایش سرعت ایجاد کند و این افزایش نسبت به روش ادغام توکن‌ها نیز بیشتر است. همچنین جدول ۴-۵ نیز نشان می‌دهد که این کاهش در مدل $Stable\ Diffusion\ v2$ با ضریب ۱/۳۱ می‌باشد که نسبت به مدل اول ضریب بهتری است.

تصاویر تولید شده توسط مدل‌های $Stable\ Diffusion\ v1.5$ و $Stable\ Diffusion\ v2$ برای سه حالت مدل پایه، با ادغام توکن و با روش پیشنهادی به ترتیب در تصویر ۴-۴ و ۴-۵ آمده‌است. نکته‌ی قابل توجه در این تصاویر این است که در هر سه دسته‌ی تصاویر تولیدشده مصنوعات یا از دست رفتن رنگ در تصویر وجود دارد و تقریبا می‌توان گفت که به جهت کیفیت تصاویر تولیدی با یکدیگر برابرند. این موضوع داده‌های جداول نتیجه را هم توجیه می‌کند که کیفیت هر سه روش تقریبا برابر اما زمان تولید در روش پیشنهادی از همه کمتر است.

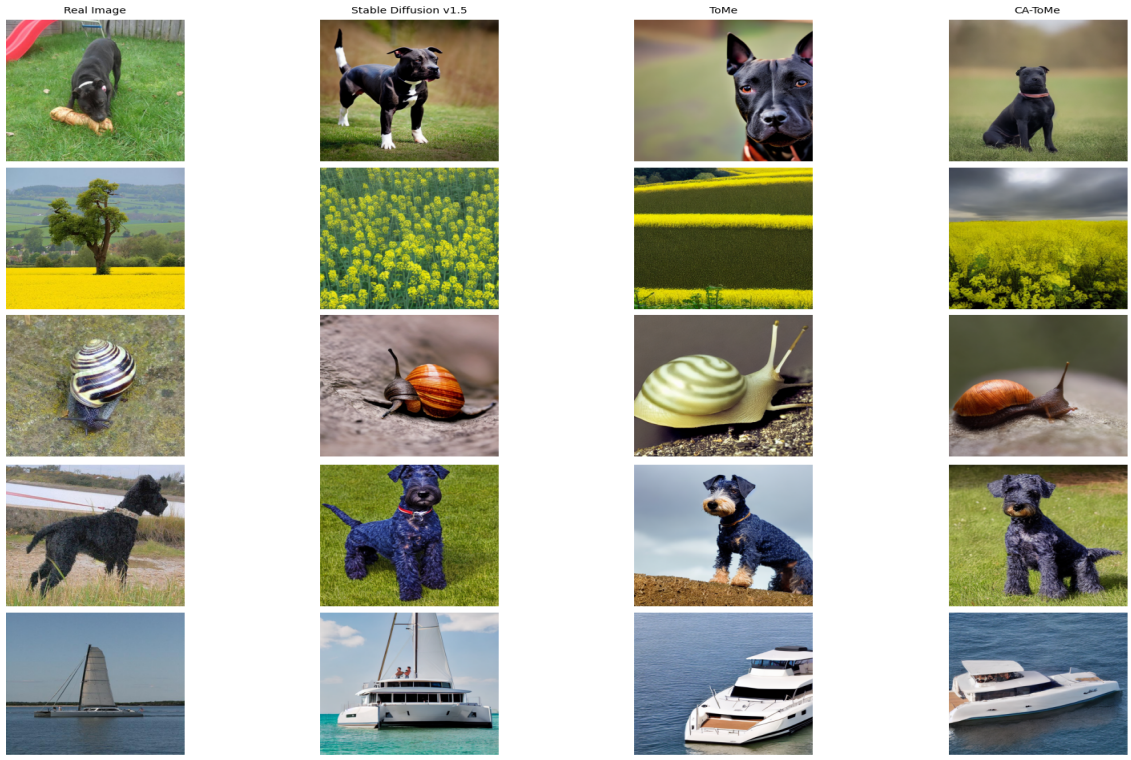
جدول ۴-۴: مقایسه‌ی مدل پایه‌ی $Stable\ Diffusion\ v1.5$و مدل ادغام توکن‌ها با روش پیشنهادی در بهترین تنظیمات آن

مدل/ روش	FID	$LPIPS$	زمان میانگین(ثانیه)
مدل $Stable\ Diffusion\ v1.5$	۳۳/۳۷	۰/۸۱۳۶	۷/۶۱± ۰/۰۰۱
روش ادغام توکن‌ها	۳۴/۱۶	۰/۸۵۱۰	۶/۳۹ ± ۰/۰۰۶
روش پیشنهادی	۳۳/۴۹	۰/۸۳۴۷	۶/۰۹ ± ۰/۰۰۱

جدول ۴-۵: مقایسه‌ی مدل پایه‌ی $Stable\ Diffusion\ v2$و مدل ادغام توکن‌ها با روش پیشنهادی در بهترین تنظیمات آن

مدل/ روش	FID	$LPIPS$	زمان میانگین(ثانیه)
مدل $Stable\ Diffusion\ v2$	۳۴/۱۴	۰/۸۰۰۹	۱۹/۰۸± ۰/۰۰۳
روش ادغام توکن‌ها	۳۴/۴۷	۰/۸۴۴۳	۱۵/۱۵ ± ۰/۰۲
روش پیشنهادی	۳۳/۲۶	۰/۸۱۵۶	۱۴/۵۰ ± ۰/۰۴

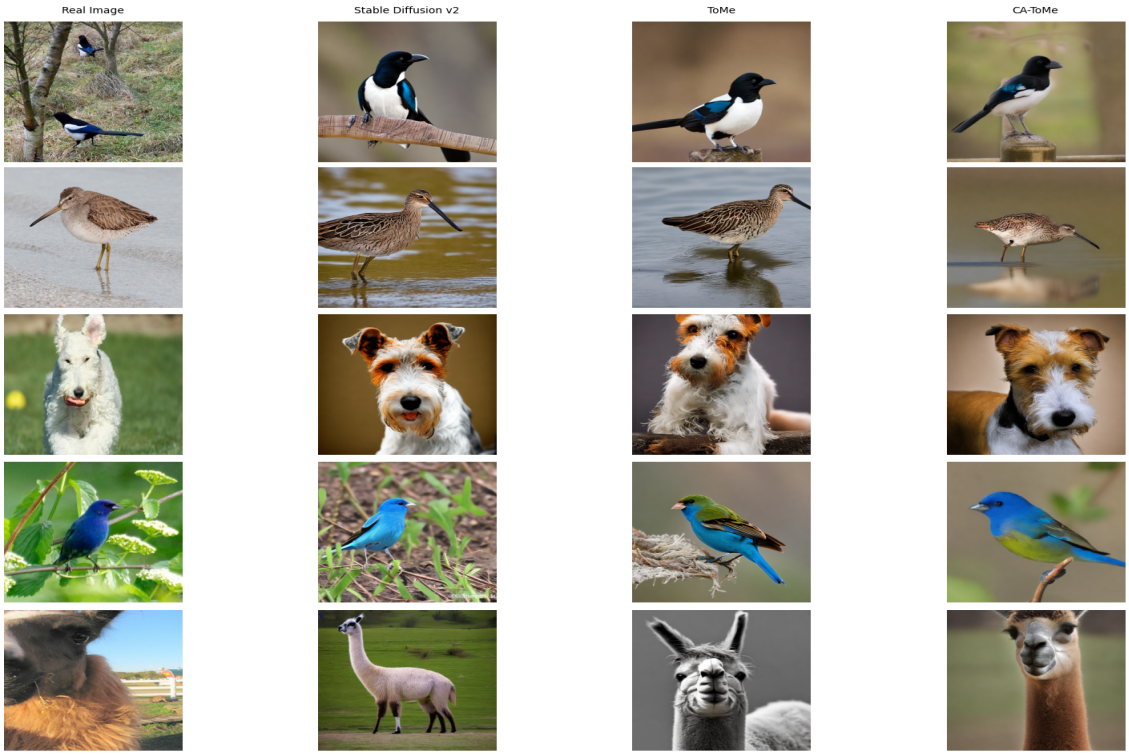
همچنین با توجه به جدول ۴-۱ در حالتی که تمام توکن‌های مبدا و مقصد را با یکدیگر ادغام کنیم، اتفاقی که در آستانه‌ی ۰/۴ می‌افتد، زمان حداقل به ۶/۰۷ ثانیه می‌رسد و در ازای این ادغام کامل دو واحد افزایش در معیار FID داریم که به معنای کاهش کیفیت است. اما روش ما بدون ادغام تمامی توکن‌ها با این مقدار زمانی تنها ۰/۰۲ ثانیه فاصله دارد در حالیکه کیفیت آن به هیچ عنوان افت نمی‌کند.



شکل ۴-۴: نمونه تصاویر تولیدی از ۵ کلاس مجموعه داده‌ی $ImageNet$ در سه حالت مدل پایه‌ی $Stable\ Diffusion\ v1.5$، همراه با ادغام توکن و همراه با روش پیشنهادی

نکته‌ی قابل نتیجه‌گیری از این موضوع این است که اگر بدون در نظر گرفتن کیفیت بخواهیم حداکثر توان این روش را برای کاهش زمان نمونه‌برداری محاسبه کنیم این عدد از حالتی به دست می‌آید که تمامی توکن‌ها را ادغام کرده باشیم یعنی در نظر گرفتن آستانه‌ی ساده‌گیرانه‌ی ۰/۴ و بنابراین روش پیشنهادی ما سعی می‌کند به این حداکثر ظرفیت کاهش زمان با این روش برسد در حالیکه کیفیت تصاویر تولیدی را ثابت نگه میدارد.

نکته‌ی دیگری که حائز اهمیت است این است که استفاده از این روش مانع استفاده از سایر روش‌های افزایش سرعت چه با کاهش تعداد گام و چه با کاهش محاسبات هر گام نمی‌گردد و می‌تواند در کنار روشی مانند استفاده از حافظه‌ی پنهان عمیق قرار گیرد تا به افزایش سرعت بیشتری برسیم.



شکل ۴-۵: نمونه تصاویر تولیدی از ۵ کلاس مجموعه داده‌ی $ImageNet$ در سه حالت مدل پایه‌ی $Stable\ Diffusion\ v2$، همراه با ادغام توکن و همراه با روش پیشنهادی

فصل پنجم

جمع‌بندی و راهکارهای آتی

در این پژوهش هدف تمرکز بر نقطه‌ی ضعف اصلی مدل‌های پخشی یعنی سرعت کم نمونه‌برداری آنها بود. در وهله‌ی اول با مطالعه و دسته‌بندی روش‌های موجود برای افزایش سرعت به یک دسته‌بندی از روش‌های کاهش تعداد گام در مقابل روش‌های کاهش محاسبات هر گام رسیدیم و سعی کردیم در هر دسته تعدادی روش را که بدون نیاز به آموزش بودند مورد مطالعه قرار دهیم. این مطالعات ما را به روشی تحت عنوان ادغام توکن‌ها رساند.
روش ادغام توکن‌ها با در نظر گرفتن اینکه هزینه‌ی محاسباتی بلوک‌های مبدل داخل شبکه‌ی یو زیاد است و اینکه خود مبدل هزینه‌ی محاسباتی درجه دو نسبت به اندازه‌ی ورودی دارد، تلاش می‌کند تا با ترکیب توکن‌های ورودی مبدل از تعداد آنها بکاهد. روش اولیه‌ی ارائه شده در این رویکرد توکن‌های ورودی را به دو دسته‌ی مبدا و مقصد تقسیم می‌کند و سپس برای هر توکن مبدا مشابه‌ترین توکن مقصد را در نظر می‌گیرد و به این ترتیب یک گراف دو بخشی می‌سازد. سپس r درصد از مشابه‌ترین توکن‌های مبدا و مقصد را با عملگر میانگین‌گیری ترکیب می‌کند.
این روش نقاط ضعفی از جمله در نظر گرفتن یک درصد ثابت برای ترکیب توکن‌ها و نیز محاسبه‌ی ماتریس مشابهت در آغاز هر مبدل داشت که خود دارای هزینه‌ی محاسباتی درجه دو است. از همین رو تلاش این پژوهش بر این بوده است که این نقاط ضعف را بررسی و برای آنها راه‌حلی ارائه دهد. مشکل اول که ثابت بودن درصد ادغام در همه‌ی بلوک‌ها و همه‌ی بازه‌های زمانی و در نتیجه‌ی آن انعطاف ناپذیری روش ادغام توکن‌ها است را تلاش کردیم تا با ارائه‌ی راه‌حلی مبنی بر در نظر گرفتن یک آستانه‌ی t و ادغام توکن‌هایی که مشابهت بیشتر از t دارند و تطبیقی کردن رویکرد ادغام توکن‌ها برطرف کنیم.
و همچنین برای کاهش حجم سربار محاسباتی روش ادغام توکن‌ها از یک روش مبتنی بر حافظه‌ی پنهان برای ذخیره‌ی محاسبات در تعدادی از گام‌ها و فراخوانی مقادیر این محاسبات از حافظه‌ی پنهان در سایر گام‌ها استفاده کردیم. این روش بر مبنای این مشاهده در قسمت‌های مختلف شبکه‌ی یو است که اشاره می‌کند این تغییرات خروجی بلوک‌های مختلف در گام‌های زمانی متوالی ناچیز و هموار است. موضوعی که با رسم فاصله‌ی جاکارد جفت‌های ادغام شده در گام‌های متوالی در ورودی مبدل نیز قابل مشاهده است.
نتایج اعمال این دو راه‌حل روش پیشنهادی ما تحت عنوان ادغام توکن تطبیقی مبتنی بر حافظه‌ی پنهان را می‌سازد. این روش بدون کاهش کیفیت می‌تواند کاهش زمانی بیشتری نسبت به روش ادغام توکن‌ها ارائه دهد. همچنین با توجه به اینکه حداکثر کاهش در روش ادغام توکن‌ها زمانی رخ می‌دهد که همه‌ی توکن‌های مبدا و مقصد را ادغام کنیم، می‌توان نشان داد که روش پیشنهادی ما این حداکثر کاهش زمانی را بدون از دست دادن کیفیت به دست می‌آورد.

در ادامه برای بهبود کارایی این روش پیشنهاداتی قابل ارائه است که به شرح زیر می‌باشند:

- استفاده از روش پیشنهادی در کنار روش استفاده از حافظه‌ی پنهان عمیق[1]: روش پیشنهادی ما تنها بر بلوک‌های با ویژگی سطح پایین و یا بلوک‌های با مقدار $DownSample = 1$ اعمال

[1] Deep Cache

می‌شود و باعث کاهش حجم محاسبات در این بلوک‌ها می‌شود. از طرفی روش استفاده از حافظه‌ی پنهان عمیق بلوک‌های میانی را در حافظه‌ی پنهان ذخیره می‌کند. ترکیب این دو روش باعث می‌شود که در تمامی بلوک‌ها یک رویکرد برای کاهش محاسبات اعمال شده باشد. البته ممکن است نیاز باشد این دو روش با روش سومی نیز برای افزایش کیفیت ترکیب شوند تا کیفیت در این فرآیند کاهش محاسبات افت نکند.

- اعمال روش پیشنهادی بر سایر معماری‌های موجود در مدل‌های پخشی: ساختار و معماری در نظر گرفته شده در سراسر این آزمایش مدل پخشی مبتنی بر شبکه‌ی یو است، حال آنکه اخیرا دسته‌های دیگری از مدل‌های پخشی از جمله مدل‌های پخشی مبتنی بر مبدل ارائه شده‌اند. بررسی عملکرد روش پیشنهادی و کلا رویکردهای کاهش توکن در این معماری‌ها موضوعی مغفول مانده از سمت محققین این حوزه است که بررسی امکان و تنظیم این روش برای سایر معماری‌ها می‌تواند یکی از راهکارهای پیش رو باشد.

- ترکیب روش پیشنهادی با سایر روش‌های عمود بر آن: در راهکار اول یک نمونه‌ی خاص از ترکیب روش پیشنهادی با روش‌های دیگر را بررسی کردیم. اما در این راهکار با توجه به قرار داشتن هر دو روش در یک دسته امکان افت کیفیت وجود داشت در حالیکه ترکیب روش پیشنهادی با سایر روش‌های کاهش گام که بدون کاهش کیفیت این عمل را انجام می‌دهند می‌تواند رویکرد دیگری باشد. استفاده از تقطیر دانش و ساخت شبکه‌ی دانش‌آموز و کاهش گام و یا استفاده از حل‌کننده‌های مختلف از جمله رویکردهایی هستند که می‌توانند عمود بر رویکرد ارائه شده پیشنهاد شوند.

- ارائه‌ی روش‌های جدید ادغام یا رفع ادغام: در معرفی انواع روش‌های ادغام در فصل دوم اشاره شد که از رویکردهایی مثل خوشه‌بندی نیز می‌توانیم برای ادغام توکن‌ها استفاده کنیم. علت عمده‌ی عدم استفاده از این روش‌ها زمانبر بودن آنها است. در روش پیشنهادی با توجه به استفاده از ایده‌ی به کارگیری حافظه‌ی پنهان می‌توان در خصوص سایر روش‌های ادغام مانند خوشه‌بندی یا رویکردهای مشابه هم فکر کرد. و در نهایت در روش پیشنهادی رفع ادغام از طریق جایگذاری توکن مقصد در تمامی توکت‌های مبدا ادغام شده با آن می‌باشد این رویکرد نیز می‌تواند بهبود پیدا کند و از روش‌های پیچیده‌تری برای ادغام استفاده شود.

کتاب‌نامه

[1] Bolya, Daniel, Fu, Cheng-Yang, Dai, Xiaoliang, Zhang, Peizhao, Feichtenhofer, Christoph, and Hoffman, Judy. Token merging: Your vit but faster. arXiv preprint arXiv:2210.09461, 2022.

[2] Bolya, Daniel and Hoffman, Judy. Token merging for fast stable diffusion. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pages 4599–4603, 2023.

[3] Cherti, Mehdi, Beaumont, Romain, Wightman, Ross, Wortsman, Mitchell, Ilharco, Gabriel, Gordon, Cade, Schuhmann, Christoph, Schmidt, Ludwig, and Jitsev, Jenia. Reproducible scaling laws for contrastive language-image learning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 2818–2829, 2023.

[4] Deng, Jia, Dong, Wei, Socher, Richard, Li, Li-Jia, Li, Kai, and Fei-Fei, Li. Imagenet: A large-scale hierarchical image database. In 2009 IEEE Conference on Computer Vision and Pattern Recognition, pages 248–255, 2009.

[5] Feller, William. Retracted chapter: On the theory of stochastic processes, with particular reference to applications. In Selected Papers I, pages 769–798. Springer, 2015.

[6] Haurum, Joakim Bruslund, Escalera, Sergio, Taylor, Graham W, and Moeslund, Thomas B. Which tokens to use? investigating token reduction in vision transformers.

In Proceedings of the IEEE/CVF International Conference on Computer Vision, pages 773–783, 2023.

[7] Heusel, Martin, Ramsauer, Hubert, Unterthiner, Thomas, Nessler, Bernhard, and Hochreiter, Sepp. Gans trained by a two time-scale update rule converge to a local nash equilibrium. Advances in neural information processing systems, 30, 2017.

[8] Hinton, Geoffrey. Distilling the knowledge in a neural network. arXiv preprint arXiv:1503.02531, 2015.

[9] Ho, Jonathan, Jain, Ajay, and Abbeel, Pieter. Denoising diffusion probabilistic models. Advances in neural information processing systems, 33:6840–6851, 2020.

[10] Kim, Minchul, Gao, Shangqian, Hsu, Yen-Chang, Shen, Yilin, and Jin, Hongxia. Token fusion: Bridging the gap between token pruning and token merging. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, pages 1383–1392, 2024.

[11] Kreis, Karsten, Gao, Ruiqi, and Vahdat, Arash. Denoising diffusion-based generative modeling: Foundations and applications, 2022.

[12] Krizhevsky, Alex, Hinton, Geoffrey, et al. Learning multiple layers of features from tiny images. 2009.

[13] Krizhevsky, Alex, Sutskever, Ilya, and Hinton, Geoffrey E. Imagenet classification with deep convolutional neural networks. Advances in neural information processing systems, 25, 2012.

[14] Li, Senmao, Hu, Taihang, Shahbaz Khan, Fahad, Li, Linxuan, Yang, Shiqi, Wang, Yaxing, Cheng, Ming-Ming, and Yang, Jian. Faster diffusion: Rethinking the role of unet encoder in diffusion models. arXiv e-prints, pages arXiv–2312, 2023.

[15] Li, Xiuyu, Liu, Yijiang, Lian, Long, Yang, Huanrui, Dong, Zhen, Kang, Daniel, Zhang, Shanghang, and Keutzer, Kurt. Q-diffusion: Quantizing diffusion models. In Proceedings of the IEEE/CVF International Conference on Computer Vision, pages 17535–17545, 2023.

[16] Lin, Tsung-Yi, Maire, Michael, Belongie, Serge, Hays, James, Perona, Pietro, Ramanan, Deva, Dollár, Piotr, and Zitnick, C Lawrence. Microsoft coco: Common objects in context. In Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13, pages 740–755. Springer, 2014.

[17] Liu, Haozhe, Zhang, Wentian, Xie, Jinheng, Faccio, Francesco, Xu, Mengmeng, Xiang, Tao, Shou, Mike Zheng, Perez-Rua, Juan-Manuel, and Schmidhuber, Jürgen. Faster diffusion via temporal attention decomposition. arXiv e-prints, pages arXiv–2404, 2024.

[18] Ma, Xinyin, Fang, Gongfan, and Wang, Xinchao. Deepcache: Accelerating diffusion models for free. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 15762–15772, 2024.

[19] Ma, Zhiyuan, Zhang, Yuzhu, Jia, Guoli, Zhao, Liangliang, Ma, Yichao, Ma, Mingjie, Liu, Gaofeng, Zhang, Kaiyan, Li, Jianjun, and Zhou, Bowen. Efficient diffusion models: A comprehensive survey from principles to practices. arXiv preprint arXiv:2410.11795, 2024.

[20] Pan, Zizheng, Zhuang, Bohan, Huang, De-An, Nie, Weili, Yu, Zhiding, Xiao, Chaowei, Cai, Jianfei, and Anandkumar, Anima. T-stitch: Accelerating sampling in pre-trained diffusion models with trajectory stitching. arXiv preprint arXiv:2402.14167, 2024.

[21] Peebles, William and Xie, Saining. Scalable diffusion models with transformers. In Proceedings of the IEEE/CVF International Conference on Computer Vision, pages 4195–4205, 2023.

[22] Radford, Alec, Kim, Jong Wook, Hallacy, Chris, Ramesh, Aditya, Goh, Gabriel, Agarwal, Sandhini, Sastry, Girish, Askell, Amanda, Mishkin, Pamela, Clark, Jack, et al. Learning transferable visual models from natural language supervision. In International conference on machine learning, pages 8748–8763. PMLR, 2021.

[23] Rombach, Robin, Blattmann, Andreas, Lorenz, Dominik, Esser, Patrick, and Ommer, Björn. High-resolution image synthesis with latent diffusion models. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pages 10684–10695, 2022.

[24] Ronneberger, Olaf, Fischer, Philipp, and Brox, Thomas. U-net: Convolutional networks for biomedical image segmentation. In Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18, pages 234–241. Springer, 2015.

[25] Salimans, Tim and Ho, Jonathan. Progressive distillation for fast sampling of diffusion models. arXiv preprint arXiv:2202.00512, 2022.

[26] Salimans, Tim, Karpathy, Andrej, Chen, Xi, and Kingma, Diederik P. Pixelcnn++: Improving the pixelcnn with discretized logistic mixture likelihood and other modifications. arXiv preprint arXiv:1701.05517, 2017.

[27] Shang, Yuzhang, Yuan, Zhihang, Xie, Bin, Wu, Bingzhe, and Yan, Yan. Post-training quantization on diffusion models. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pages 1972–1981, 2023.

[28] Simonyan, Karen. Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556, 2014.

[29] Song, Jiaming, Meng, Chenlin, and Ermon, Stefano. Denoising diffusion implicit models. arXiv preprint arXiv:2010.02502, 2020.

[30] Song, Yang and Ermon, Stefano. Generative modeling by estimating gradients of the data distribution. Advances in neural information processing systems, 32, 2019.

[31] Song, Yang and Ermon, Stefano. Improved techniques for training score-based generative models. Advances in neural information processing systems, 33:12438–12448, 2020.

[32] Song, Yang, Sohl-Dickstein, Jascha, Kingma, Diederik P, Kumar, Abhishek, Ermon, Stefano, and Poole, Ben. Score-based generative modeling through stochastic differential equations. arXiv preprint arXiv:2011.13456, 2020.

[33] Wimbauer, Felix, Wu, Bichen, Schoenfeld, Edgar, Dai, Xiaoliang, Hou, Ji, He, Zijian, Sanakoyeu, Artsiom, Zhang, Peizhao, Tsai, Sam, Kohler, Jonas, et al. Cache me if you can: Accelerating diffusion models through block caching. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 6211–6220, 2024.

[34] Xiao, Zhisheng, Kreis, Karsten, and Vahdat, Arash. Tackling the generative learning trilemma with denoising diffusion gans. arXiv preprint arXiv:2112.07804, 2021.

[35] Zhang, Richard, Isola, Phillip, Efros, Alexei A, Shechtman, Eli, and Wang, Oliver. The unreasonable effectiveness of deep features as a perceptual metric. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 586–595, 2018.

واژه‌نامه‌ی فارسی به انگلیسی

ا

افزایش کیفیت . . . . Fidelity Improving

انتشار کدگذار . . . Encoder Propagation

افزونگی . . . . . Redundancy

ادغام توکن‌ها . . . . . Token Merging

ادغام سخت . . . . . Hard-merging

ادغام نرم . . . . . Soft-merging

ادغام توکن . . . . . ToMe

اعداد ممیز شناور . . . . . Floating-point

اشیای مشترک مایکروسافت در متن Microsoft Common Objects in Context

آموزش آگاه از کوانتیزاسیون Quantization-aware Training

ب

برنامه‌ی واریانس . . . Variance Schedule

بهترین فناوری موجود . State of the Art

برنامه‌ریزی معنایی . Semantic Planning

پ

پویا . . . . . Dynamic

ت

توضیح تصویر . . . . . Caption

تابع محدودسازی . . . Clipping function

تقطیر دانش . . Knowledge Distillation

تقطیر دانش مبتنی بر توزیع Distribution-based KD

تقطیر دانش مبتنی بر مسیر . Trajectory-based KD

تقطیر دانش تقابلی . . . Adversarial KD

تمایزگر . . . . . Discriminator

تولید کننده . . . . . Generator

ث

ثابت . . . . . Static

ح

حد پایین مشاهده . . . . Evidence lower bound

حد تغییراتی . . . . . Variational bound

حافظه‌ی پنهان عمیق . . . . Deep Cache

خ

خودکدگذارهای تغییراتی . . . Variational autoencoders

خطی بودن تابعی . Functional Linearity

د

درست‌نمایی . . . . . Likelihood

درون‌یابی خطی حداکثر نرم . Maximum Norm Linear Interpolation

ذ

ذخیره‌سازی در حافظه‌ی پنهان . Caching

ر

رفع ادغام . . . . . Unmerging

روش‌های مبتنی بر آموزش . . . Training-based Methods

روش‌های بدون نیاز به آموزش . Training-Free Methods

رویکرد گیت . . . . . TGate

ز

زنجیره‌ی مارکوف . . . . . Markov Chain

ش

شباهت کسینوسی . . Cosine Similarity

شبکه‌های مولد تقابلی . . . . . Generative adversarial networks

شبکه‌ی یو نویززدا . . . Denoising U-net

شبکه‌ی از پیش آموزش‌دیده Pre-trained Network

شباهت ادراکی یادگرفته شده‌ی قطعات تصویر . . Learned Perceptual Image Patch Similarity

شبکه‌ی کلمات . . . . . WordNet

غ

غیر یکنواخت . . . . . Non-uniform

ف

فرآیند رو به جلو . . . . Forward process

فرآیند معکوس . . . . . Reverse Process

فشرده‌سازی شبکه . . . . . Network Compression

فعال‌سازی . . . . . Activation

فرمان . . . . . Prompt

فاصله‌ی اکتسابی فرچت . . . . . Frechet Inception Distance

ک

کوانتیزاسیون . . . . . Quantization

کاهش توکن . . . . . Token Reduction

کوانتیزاسیون پس از آموزش Post-training Quantization

کوانتیزاسیون مدل‌های پخشی . . . . . Q-diffusion

گ

گلوگاه . . . . . Bottleneck

ل

لگاریتم درست‌نمایی . . . Log-likelihood

م

مبدل . . . . . Transformer

مبدل بینایی Vision Transformer (VIT)

مجموعه‌ی هم‌معنی . . . . . Synonym Set

مجموعه داده‌ی کالیبراسیون Calibration Dataset

مدل‌های مولد . . . . Generative Models

مدل‌های پخشی . . . . Diffusion Models

مدل‌های پخشی مبتنی بر امتیاز . Score-based Diffusion Models

مدل‌های پخشی نویززدای احتمالاتی Denoising Diffusion Probabilistic Models

مدل‌های مبتنی بر معادلات دیفرانسیل تصادفی . Stochastic Differential Equation based Models

مدل‌های پخشی ضمنی نویززدا Denoising Diffusion Implicit Models (DDIM)

مدل‌های متغیر پنهان . . Latent variable models

مسیر رو به جلو . . . . . Forward Path

مسیر رو به عقب . . . . . Backward Path

مبدا . . . . . Source

مقصد . . . . . Destination

مدل پخشی سریع‌تر . . Faster Diffusion

ن

ناهنجاری . . . . . Anomaly

نرم . . . . . Norm

نمونه‌برداری افزایشی . . . . . Upsampling

نقشه‌ی حرارتی . . . . . Heatmap

نمونه‌برداری کاهشی . . . Downsampling

نرم فرینیوس . . . . . Frobenius Norm

ه

همسانگرد . . . . . Isotropic

هسته‌ی گذر . . . . . Transition kernel

هرس توکن . . . . . Token Pruning

همجوشی توکن . . . . . Token Fusion

ی

یادگیری نیمه‌نظارتی . Semi-supervised learning

یکنواخت . . . . . Uniform

واژه‌نامه‌ی انگلیسی به فارسی

A

Activation . . . . . فعال‌سازی

Adversarial KD . . . تقطیر دانش تقابلی

Anomaly . . . . . ناهنجاری

B

Backward Path . . . . . مسیر رو به عقب

Bottleneck . . . . . گلوگاه

C

Calibration Dataset مجموعه داده‌ی کالیبراسیون

Caching . ذخیره‌سازی در حافظه‌ی پنهان

Caption . . . . . توضیح تصویر

Clipping function . . . تابع محدودسازی

Cosine Similarity . . شباهت کسینوسی

D

Deep Cache . . . . حافظه‌ی پنهان عمیق

Denoising Diffusion Implicit Models (DDIM) مدل‌های پخشی ضمنی نویززدا

Denoising Diffusion Probabilistic Models مدل‌های پخشی نویززدای احتمالاتی

Denoising U-net . . . شبکه‌ی یو نویززدا

Discriminator . . . . . تمایزگر

Distribution-based KD تقطیر دانش مبتنی بر توزیع

Diffusion models . . . . مدل‌های پخشی

Downsampling . . . نمونه‌برداری کاهشی

Dynamic . . . . . پویا

E

Encoder Propagation . . . انتشار کدگذار

Evidence lower bound . . . . حد پایین مشاهده

F

Faster Diffusion . . مدل پخشی سریع‌تر

Fidelity Improving . . . . افزایش کیفیت

درست‌نمایی . . . . . Likelihood

لگاریتم درست‌نمایی . . . Log-likelihood

M

زنجیره‌ی مارکوف . . . . . Markov Chain

درون‌یابی خطی حداکثر نرم . Maximum Norm Linear Interpolation

اشیای مشترک مایکروسافت در متن Microsoft Common Objects in Context

N

فرمان‌های منفی . . . Negative Prompts

فشرده‌سازی شبکه . . . . . Network Compression

نرم . . . . . Norm

غیر یکنواخت . . . . . Non-uniform

P

کوانتیزاسیون پس از آموزش Post-training Quantization

شبکه‌ی از پیش آموزش‌دیده Pre-trained Network

فرمان . . . . . Prompt

Q

کوانتیزاسیون مدل‌های پخشی . . . . . Q-diffusion

کوانتیزاسیون . . . . . Quantization

اعداد ممیز شناور . . . . . Floating-point

مسیر رو به جلو . . . . . Forward Path

فرآیند رو به جلو . . . . Forward process

فاصله‌ی اکتسابی فرچت . . . . . Frechet Inception Distance

نرم فرینیوس . . . . . Frobenius Norm

خطی بودن تابعی . Functional Linearity

G

شبکه‌های مولد تقابلی . . . . . Generative adversarial networks

مدل‌های مولد . . . . Generative Models

تولید کننده . . . . . Generator

H

ادغام سخت . . . . . Hard-merging

نقشه‌ی حرارتی . . . . . Heatmap

I

همسانگرد . . . . . Isotropic

K

تقطیر دانش . . Knowledge Distillation

L

مدل‌های متغیر پنهان . . Latent variable models

شباهت ادراکی یادگرفته شده‌ی قطعات تصویر . . Learned Perceptual Image Patch Similarity

ادغام توکن‌ها . . . . . Token Merging

هرس توکن . . . . . Token Pruning

کاهش توکن . . . . . Token Reduction

تقطیر دانش مبتنی بر مسیر . Trajectory-based KD

روش‌های مبتنی بر آموزش . . . Training-based Methods

روش‌های بدون نیاز به آموزش . Training-Free Methods

مبدل . . . . . Transformer

هسته‌ی گذر . . . . . Transition kernel

U

یکنواخت . . . . . Uniform

رفع ادغام . . . . . Unmerging

نمونه‌برداری افزایشی . . . . . Upsampling

V

برنامه‌ی واریانس . . . Variance Schedule

خودکدگذارهای تغییراتی . . . Variational autoencoders

حد تغییراتی . . . . . Variational bound

مبدل بینایی Vision Transformer (VIT)

W

شبکه‌ی کلمات . . . . . WordNet

آموزش آگاه از کوانتیزاسیون Quantization-aware Training

R

افزونگی . . . . . Redundancy

فرآیند معکوس . . . . . Reverse Process

S

مدل‌های پخشی مبتنی بر امتیاز . Score-based Diffusion Models

برنامه‌ریزی معنایی . Semantic Planning

یادگیری نیمه‌نظارتی . Semi-supervised learning

ادغام نرم . . . . . Soft-merging

مبدا . . . . . Source

بهترین فناوری موجود . State of the Art

ثابت . . . . . Static

مدل‌های مبتنی بر معادلات دیفرانسیل تصادفی . Stochastic Differential Equation based Models

مجموعه‌ی هم‌معنی . . . . . Synonym Set

T

رویکرد گیت . . . . . TGate

ادغام توکن . . . . . ToMe

همجوشی توکن . . . . . Token Fusion

ادغام توکن . . . . . Token Merging

Amirkabir University of Technology
(Tehran Polytechnic)

Department of Computer Engineering

M. Sc. Thesis

Speed Improvements on diffusion models

By

Atiyeh Gh. Moghaddam

Supervisor

Dr. Ahmad NickAbadi

Feburary 2025