دانشگاه صنعتی شریف
دانشکده مهندسی کامپیوتر

پایان‌نامه کارشناسی ارشد
مهندسی کامپیوتر گرایش هوش مصنوعی و رباتیک

مدیریت فراموشی در یادگیری پیوسته با کمک روش‌های یادگیری ژرف

نگارش

فاطمه رائیجیان

استاد راهنما

دکتر حمید بیگی

شهریور ۱۴۰۵

بسم الله الرحمن الرحیم

به نام خدا

دانشگاه صنعتی شریف

دانشکده مهندسی کامپیوتر

پایان‌نامه کارشناسی ارشد

این پایان‌نامه به عنوان تحقق بخشی از شرایط دریافت درجه کارشناسی ارشد است.

عنوان: مدیریت فراموشی در یادگیری پیوسته با کمک روش‌های یادگیری ژرف

نگارش: فاطمه رائیجیان

کمیته ممتحنین

استاد راهنما: دکتر حمید بیگی امضاء:

استاد ممتحن داخلی: امضاء:

استاد ممتحن خارجی: امضاء:

تاریخ:

اظهارنامه

(اصالت متن و محتوای پایان‌نامه کارشناسی ارشد)

عنوان پایان‌نامه: مدیریت فراموشی در یادگیری پیوسته با کمک روش‌های یادگیری ژرف

استاد راهنما: دکتر حمید بیگی

این‌جانب فاطمه رائیجیان اظهار می‌دارم:

۱. متن و نتایج علمی ارائه شده در این پایان‌نامه اصیل بوده و زیرنظر استادان نام‌برده شده در بالا تهیه شده است.

۲. متن پایان‌نامه به این صورت در هیچ جای دیگری منتشر نشده است.

۳. متن و نتایج مندرج در این پایان‌نامه، حاصل تحقیقات این‌جانب به عنوان دانشجوی کارشناسی ارشد دانشگاه صنعتی شریف است.

۴. کلیه مطالبی که از منابع دیگر در این پایان‌نامه مورد استفاده قرار گرفته، با ذکر مرجع مشخص شده است.

نگارنده: فاطمه رائیجیان

تاریخ:

امضاء:

نتایج تحقیقات مندرج در این پایان‌نامه و دستاوردهای مادی و معنوی ناشی از آن (شامل فرمول‌ها، توابع کتابخانه‌ای، نرم‌افزارها، سخت‌افزارها و مواردی که قابلیت ثبت اختراع دارد) متعلق به دانشگاه صنعتی شریف است. هیچ شخصیت حقیقی یا حقوقی بدون کسب اجازه از دانشگاه صنعتی شریف حق فروش و ادعای مالکیت مادی یا معنوی بر آن یا ثبت اختراع از آن را ندارد. همچنین، کلیه حقوق مربوط به چاپ، تکثیر، نسخه‌برداری، ترجمه، اقتباس و نظائر آن در محیط‌های مختلف اعم از الکترونیکی، مجازی یا فیزیکی برای دانشگاه صنعتی شریف محفوظ است. نقل مطلب با ذکر ماخذ بلامانع است.

استاد راهنما: دکتر حمید بیگی

تاریخ:

امضاء:

نگارنده: فاطمه رائیجیان

تاریخ:

امضاء:

سپاس‌گزاری

سپاسگزارم از استاد راهنمای گرامی، جناب آقای دکتر حمید بیگی، برای راهنمایی‌های علمی، دقت نظر و حمایت مستمر در مسیر انجام این پژوهش. همچنین از خانواده‌ی عزیزم برای همراهی، صبوری و پشتیبانی بی‌دریغشان در تمام مراحل تحصیل و نگارش این پایان‌نامه قدردانی می‌کنم.

چکیده

سامانه‌های یادگیری پیوسته[1] مستقر باید به درخواست حذف دانش پاسخ دهند. در تنظیم افزایشی‌وظیفه[2]، فراموشی انتخابی[3] یعنی سرکوب یک وظیفه‌ی مشخص با حفظ عملکرد روی وظایف نگه‌داری‌شده. مانع اصلی هم‌پوشانی پارامتری[4] است: وقتی وظایف پارامتر مشترک دارند، ویرایش وزن برای یک وظیفه به وظایف دیگر آسیب می‌زند. چارچوب PALL دانش هر وظیفه را در زیرشبکه‌های ذخیره‌شده پیمانه‌ای[5] می‌کند، اما مختصه‌های مشترک را بر حسب حساسیت وظایف نگه‌داری‌شده تفکیک نمی‌کند.

این پایان‌نامه EPALL را معرفی می‌کند: مختصه‌های مشترک میان زیرشبکه‌ی هدف و اجتماع زیرشبکه‌های فعال با بزرگی گرادیان وظایف نگه‌داری‌شده رتبه‌بندی می‌شوند و حساس‌ترین زیرمجموعه‌ی آن‌ها در طول ترمیم با یک جمله‌ی لنگر[6] $\ell_۲$ مهار می‌شود. مسیر کارآمد پارامتری[7] PALL-Adapter نیز به‌عنوان گزینه‌ای اکتشافی ممیزی می‌شود.

در مقایسه‌ای میان یازده روش و روی هشت بذر روی Split-CIFAR، EPALL دقت وظیفه‌ی فراموش‌شده را به سطح تصادفی می‌رساند (۰/۴۸۶۳ و ۰/۰۹۵۹) و دقت نگه‌داری را در حد CLPU نگه می‌دارد، با حالت مقیم ۱/۶ تا ۲/۲ برابر کوچک‌تر. اختلاف جفتی دقت نسبت به PALL روی هر دو مجموعه‌داده معنادار می‌ماند ($p_{\text{raw}} = ۰/۰۰۷۸$، $p_{\text{Holm}} = ۰/۰۴۶۹$)؛ بیشترین افت روی CIFAR-100 از ۰/۰۲۲۲ به ۰/۰۰۵۶ می‌رسد، ولی روی CIFAR-10 پس از اصلاح چندگانگی معنادار نمی‌ماند. در محکی ۳۰۰ اجرایی، بیشترین افتِ PALL با رشد موقعیت درخواست بالا می‌رود و EPALL مسطح می‌ماند؛ و در جاروب پراکندگی، سود حفاظت با هم‌پوشانی اندازه‌گیری‌شده همبستگی رتبه‌ای مثبت و معنادار دارد.

کنترل‌های مؤلفه‌ای دامنه‌ی ادعا را باریک می‌کنند: تنها رتبه‌بندی حساسیت با بودجه‌ی ثابت پشتیبانی می‌شود، و سرکوب در مسیر آداپتر از بازنشانی می‌آید نه از ماسک نرم. آماره‌ی تفکیک‌پذیری حمله‌ی آستانه نشان می‌دهد فراموشی نمونه‌های هدف را تفکیک‌پذیرتر می‌کند؛ هیچ نتیجه‌ای حذف دقیق یا گواهی‌شده را اثبات نمی‌کند.

کلیدواژه‌ها: یادگیری پیوسته، یادگیری‌زدایی ماشینی، فراموشی انتخابی، هم‌پوشانی پارامتری، زیرشبکه‌های

[1] Continual Learning
[2] Task-Incremental
[3] Selective Forgetting
[4] Parameter Overlap
[5] Modular
[6] Anchor
[7] Parameter-Efficient

محافظت‌شده، یادگیری کارآمد پارامتری

فهرست مطالب

۱ مقدمه ۱

۱-۱ زمینه و انگیزه ۱

۱-۲ تعریف مسئله ۳

۱-۳ چالش اصلی: هم‌پوشانی پارامتری ۳

۱-۴ پرسش‌های پژوهش ۴

۱-۵ حدود و دامنه‌ی ادعاها ۵

۱-۶ مشارکت‌های پژوهش ۶

۱-۷ معیارهای موفقیت ۷

۱-۸ ساختار پایان‌نامه ۷

۲ مفاهیم پایه و مرور کارهای پیشین ۹

۲-۱ مفاهیم پایه ۹

۲-۱-۱ یادگیری بانظارت و بهینه‌سازی گرادیانی ۹

۲-۱-۲ سنجه‌های اهمیت پارامتر ۱۰

۲-۱-۳ معماری‌های استفاده‌شده ۱۰

۲-۱-۴ حافظه‌ی محدود و بازپخش ۱۱

۲-۲ یادگیری پیوسته ۱۱

۲-۲-۱ سه سناریو ۱۱

۲-۲-۲ روش‌های مبتنی بر تنظیم ۱۲
۲-۲-۳ روش‌های مبتنی بر بازپخش ۱۳
۲-۲-۴ تصویر گرادیان ۱۴
۲-۲-۵ جداسازی پارامتر و زیرشبکه‌های پراکنده ۱۴
۲-۳ یادگیری‌زدایی ماشینی ۱۵
۲-۳-۱ فراموشی دقیق و بازآموزی ۱۵
۲-۳-۲ فراموشی تقریبی ۱۶
۲-۳-۳ تضمین‌های صوری و مرزهای آن ۱۶
۲-۳-۴ یادگیری‌زدایی در بستر یادگیری پیوسته ۱۷
۲-۴ یادگیری کارآمد پارامتری ۱۸
۲-۵ ارزیابی فراموشی و حریم خصوصی ۱۹
۲-۵-۱ معیارهای رفتاری ۱۹
۲-۵-۲ حمله‌ی استنتاج عضویت ۱۹
۲-۵-۳ مرجع بازآموزی و ممیزی صوری ۲۰
۲-۶ مقایسه‌ی مفهومی و جایگاه این پژوهش ۲۰

۳ روش پیشنهادی ۲۳
۳-۱ صورت‌بندی مسئله ۲۳
۳-۱-۱ نمادگذاری ۲۳
۳-۱-۲ هدف مسئله ۲۴
۳-۱-۳ سازوکار ارثی PALL ۲۴
۳-۲ محدودیت فراموشی بی‌قید ۲۵
۳-۳ روش پیشنهادی EPALL ۲۶
۳-۳-۱ هم‌پوشانی ساختاری ۲۶
۳-۳-۲ رتبه‌بندی حساسیت و انتخاب بودجه ۲۷

۳-۳-۳ ترمیم لنگردار ۲۷
۳-۳-۴ الگوریتم کامل ۲۸
۳-۳-۵ کنترل‌های مؤلفه‌ای ۳۰
۳-۳-۶ تحلیل پیچیدگی ۳۰
۳-۴ مسیر اکتشافی PALL-Adapter ۳۱
۳-۴-۱ معماری ۳۱
۳-۴-۲ فراموشی سه‌فازی ۳۲
۳-۴-۳ دامنه‌ی واقعی ماسک نرم ۳۳
۳-۵ تحلیل نظری با دامنه‌ی محدود ۳۳
۳-۵-۱ نمادگذاری و افراز ۳۴
۳-۵-۲ فرض‌ها ۳۴
۳-۵-۳ کران بالای افت زیرفاز و یکنوایی آن نسبت به p ۳۵
۳-۵-۴ چرا این کران، افت انتهابه‌انتها را کران نمی‌زند ۳۷
۳-۶ پروتکل دسترسی به داده ۳۸
۳-۷ دیدگاه یکپارچه و جمع‌بندی فصل ۳۹

۴ آزمایش‌ها و نتایج **۴۰**
۴-۱ طراحی آزمایش ۴۰
۴-۱-۱ محک‌ها و افراز وظیفه‌ها ۴۰
۴-۱-۲ سه پروتکل مجزا ۴۱
۴-۱-۳ معماری‌ها و ابرپارامترها ۴۱
۴-۱-۴ بذرها و سخت‌افزار ۴۲
۴-۲ معیارهای ارزیابی ۴۲
۴-۳ مقایسه‌ی اصلی استاندارد ۴۳
۴-۳-۱ خوانش نتایج اصلی ۴۵

۴-۳-۲ تحلیل جفتی و آزمون معناداری ۵۰
۴-۴ محک موقعیت درخواست ۵۲
۴-۵ جاروب پراکندگی و هم‌پوشانی اندازه‌گیری‌شده ۵۵
۴-۶ انتساب سازوکار EPALL ۵۷
۴-۷ رژیم پیش‌آموزش‌دیده‌ی PEFT ۶۰
۴-۸ انتساب مؤلفه‌های PALL-Adapter: یک نتیجه‌ی منفی ۶۱
۴-۸-۱ ماسک نرم پیوسته: جاروب شدت ۶۳
۴-۹ نسخه‌های PALL زیر پروتکل هم‌پوشانی‌سنگین ۶۵
۴-۱۰ هزینه: دامنه‌ی ترمیم و حالت مقیم ۶۷
۴-۱۰-۱ نسبت اسمی دامنه‌ی ترمیم ۶۷
۴-۱۰-۲ حسابداری حالت مقیم ۶۸
۴-۱۱ ممیزی حریم خصوصی و ماندگاری ۶۸
۴-۱۱-۱ حمله‌ی استنتاج عضویت اصلاح‌شده ۶۸
۴-۱۱-۲ آماره‌ی تفکیک‌پذیری حمله‌ی آستانه ۶۹
۴-۱۱-۳ نسبت هنجار گرادیان ۷۲
۴-۱۱-۴ مرجع بازآموزی هم‌روش ۷۳
۴-۱۱-۵ ماندگاری سرکوب ۷۳
۴-۱۲ مطالعات حساسیت ۷۶
۴-۱۲-۱ انتخاب لنگر ۷۶
۴-۱۲-۲ معیار اهمیت: بزرگی گرادیان در برابر انرژی تعارض ۷۷
۴-۱۲-۳ انتقال میان‌معماری با ViT ۷۸
۴-۱۲-۴ عرض گلوگاه آداپتر ۷۹
۴-۱۳ تهدیدهای اعتبار ۷۹
۴-۱۴ جمع‌بندی یافته‌ها ۸۱

۵ جمع‌بندی و کارهای آینده ۸۳
۵-۱ خلاصه‌ی کار انجام‌شده ۸۳
۵-۲ پاسخ به پرسش‌های پژوهش ۸۴
۵-۳ مشارکت‌های نهایی ۸۵
۵-۴ محدودیت‌ها ۸۶
۵-۵ کارهای آینده ۸۸
۵-۶ سخن پایانی ۸۹

مراجع ۹۰

واژه‌نامه ۹۸

پیوست ۱۰۱

فهرست جداول

۲-۱ مقایسه‌ی مفهومی روش‌های مرتبط ۲۱

۴-۱ مقایسه‌ی هم‌تراز روی Split-CIFAR-10 . . . . . ۴۴

۴-۲ مقایسه‌ی هم‌تراز روی Split-CIFAR-100 . . . . . ۴۵

۴-۳ اختلاف‌های جفتی EPALL و PALL اصلی ۵۰

۴-۴ آزمون‌های دقیق معناداری جفتی با اصلاح Holm ۵۱

۴-۵ بیشترین افت در پنج درجه‌ی موقعیت درخواست ۵۳

۴-۶ رگرسیون‌های تجمیعی روی هم‌پوشانی اندازه‌گیری‌شده ۵۵

۴-۷ جاروب پراکندگی و هم‌پوشانی اندازه‌گیری‌شده ۵۶

۴-۸ کنترل‌های مستقیم سازوکار EPALL ۵۸

۴-۹ اختلاف‌های جفتی کنترل‌های EPALL ۵۸

۴-۱۰ رژیم پیش‌آموزش‌دیده‌ی PEFT ۶۱

۴-۱۱ کنترل‌های مؤلفه‌ای PALL-Adapter ۶۲

۴-۱۲ جاروب شدت ماسک نرم پیوسته ۶۴

۴-۱۳ نسخه‌های PALL زیر پروتکل هم‌پوشانی‌سنگین ۶۶

۴-۱۴ حسابداری حالت تنسوری مقیم ۶۸

۴-۱۵ حمله‌ی استنتاج عضویت افزایش‌داده‌ی هم‌تراز ۶۹

۴-۱۶ آماره‌ی تفکیک‌پذیری حمله‌ی آستانه ۷۱

۴-۱۷ نسبت هنجار گرادیان فراموشی به نگه‌داری ۷۲

۴-۱۸ ممیزی مرجع بازآموزی هم‌روش ۷۳
۴-۱۹ ممیزی ماندگاری سرکوب وظیفه‌ی فراموش‌شده ۷۴
۴-۲۰ مطالعه‌ی انتخاب لنگر روی هشت بذر ۷۶
۴-۲۱ مطالعه‌ی معیار اهمیت ۷۸
۴-۲۲ انتقال میان‌معماری روی بدنه‌ی ViT-T/8 . . . . . ۷۸
۴-۲۳ جاروب عرض گلوگاه آداپتر ۷۹

۵-۱ پیکربندی کامل آزمایش‌ها ۱۰۱

فهرست تصاویر

۱-۱ چرخه‌ی عملیاتی فراموشی انتخابی ۲

۱-۲ شهود هم‌پوشانی پارامتری ۴

۲-۱ رده‌بندی مفهومی روش‌های مرتبط ۲۱

۳-۱ سازوکار EPALL ۲۶

۳-۲ معماری PALL-Adapter ۳۲

۳-۳ یکنوایی کران بالای افت زیرفاز نسبت به شدت محافظت ۳۷

۴-۱ دقت نهایی به تفکیک مجموعه‌داده و رژیم ۴۷

۴-۲ بیشترین افت به تفکیک مجموعه‌داده و رژیم ۴۸

۴-۳ دقت وظیفه‌ی حذف‌شده به تفکیک مجموعه‌داده و رژیم ۴۹

۴-۴ پاسخ بیشترین افت به موقعیت درخواست ۵۴

۴-۵ مصالحه‌ی دامنه‌ی ترمیم و کیفیت ۶۷

۴-۶ ترکیب حالت مقیم ۶۹

۴-۷ AUC حمله‌ی عضویت پیش و پس از درخواست ۷۰

۴-۸ مسیر فاصله تا شانس وظیفه‌ی فراموش‌شده پس از حذف ۷۵

۴-۹ جاروب عرض گلوگاه آداپترها ۸۰

۴-۱۰ نقشه‌ی حرارتی دقت یک اجرای نماینده ۸۰

فصل ۱

مقدمه

۱-۱ زمینه و انگیزه

شبکه‌های عصبی عمیق در دو دهه‌ی گذشته به ابزار غالب یادگیری بازنمایی از داده‌های ادراکی تبدیل شده‌اند [۱، ۲]. اما الگوی متعارف آموزش این شبکه‌ها یک فرض ساده‌کننده‌ی مهم دارد: کل مجموعه‌ی داده از پیش موجود است، به‌طور تصادفی برزده می‌شود و مدل با گذرهای متعدد روی همین توزیع ثابت آموزش می‌بیند. سامانه‌های واقعی تقریبا هیچ‌گاه در چنین شرایطی کار نمی‌کنند. داده به‌صورت جریانی می‌رسد، مفاهیم تازه پس از استقرار مدل ظاهر می‌شوند و آموزش دوباره‌ی مدل از صفر پس از هر تغییر، از نظر محاسباتی و اقتصادی صرفه ندارد.

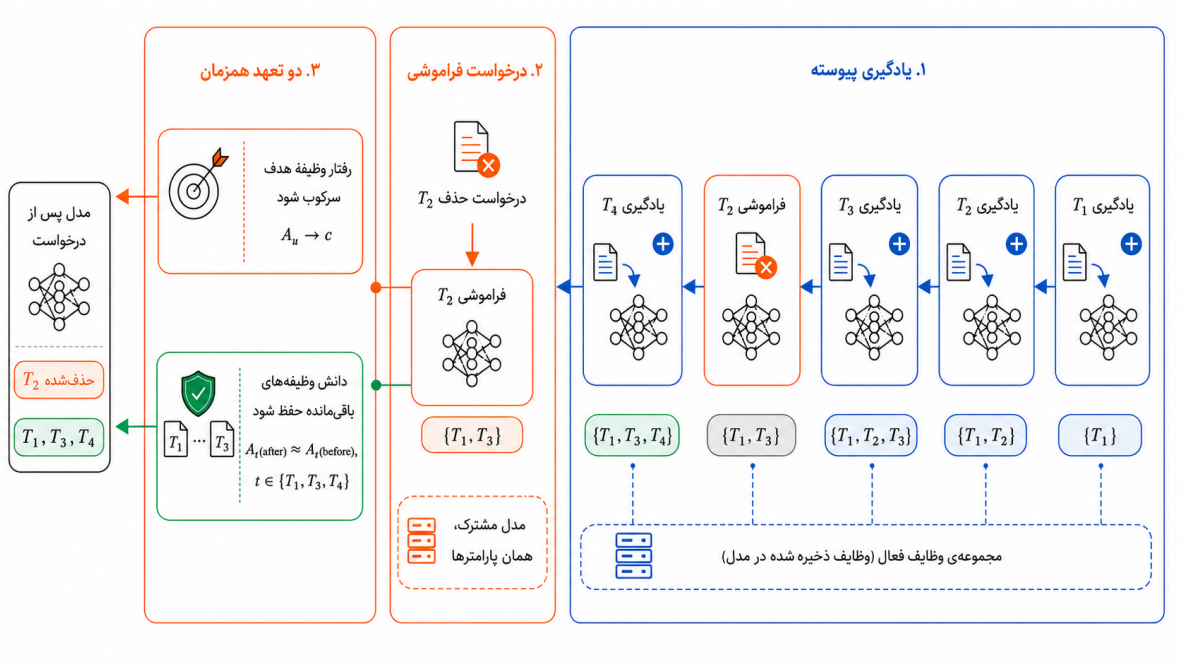
یادگیری پیوسته[1] یا یادگیری مادام‌العمر[2] پاسخ پژوهشی به این وضعیت است: مدل باید دنباله‌ای از وظیفه‌ها را یکی پس از دیگری فرا بگیرد و دانش پیشین را از دست ندهد [۳، ۴، ۵]. دشواری بنیادی این حوزه فراموشی فاجعه‌بار[3] است: چون گام‌های گرادیانی وظیفه‌ی جدید همان پارامترهایی را جابه‌جا می‌کنند که وظیفه‌های پیشین به آن‌ها تکیه کرده‌اند، عملکرد روی وظیفه‌های قدیمی به‌سرعت افت می‌کند [۶، ۷]. بخش عمده‌ی ادبیات یادگیری پیوسته دقیقا حول همین محور سازمان یافته است: چگونه می‌توان دانش انباشته را در برابر به‌روزرسانی‌های آینده حفظ کرد.

هم‌زمان، فشار حقوقی و اجتماعی مسئله‌ای تقریبا معکوس را روی میز گذاشته است. مقررات حفاظت از داده مانند مقررات عمومی حفاظت از داده‌ی اتحادیه‌ی اروپا[4] [۸] و قانون حریم خصوصی مصرف‌کننده‌ی

[1] Continual Learning
[2] Lifelong Learning
[3] Catastrophic Forgetting
[4] General Data Protection Regulation (GDPR)

کالیفرنیا[۵] [۹] برای افراد حق فراموش‌شدن[۶] را به رسمیت می‌شناسند. اگر داده‌ی یک فرد یا یک دامنه‌ی مشخص باید از سامانه حذف شود، حذف رکورد از پایگاه داده کافی نیست؛ مدلی که روی آن داده آموزش دیده است هم اثر آن را در وزن‌های خود نگه می‌دارد و این اثر با حمله‌های استنتاجی قابل بهره‌برداری است [۱۰، ۱۱]. یادگیری‌زدایی ماشینی[۷] به‌عنوان یک زمینه‌ی پژوهشی مستقل شکل گرفت تا همین شکاف را پر کند [۱۲، ۱۳].

تقاطع این دو خط، مسئله‌ی این پایان‌نامه است. یک سامانه‌ی یادگیری پیوسته‌ی مستقر، هم باید در برابر فراموشی ناخواسته مقاوم باشد و هم باید بتواند فراموشی خواسته‌شده را به‌درستی اجرا کند. این دو نیاز در سطح سازوکار با یکدیگر در تنش‌اند: هر ابزاری که برای تثبیت دانش گذشته ساخته شده است، حذف عامدانه‌ی بخشی از آن دانش را دشوارتر می‌کند، و هر عملیات حذفی که بی‌قید اجرا شود، همان دانشی را که باید حفظ می‌شد از بین می‌برد. شکل ۱-۱ این چرخه‌ی عملیاتی را نشان می‌دهد.



شکل ۱-۱: چرخه‌ی عملیاتی یک سامانه‌ی یادگیری پیوسته‌ی پاسخ‌گو به درخواست حذف. مدل وظیفه‌ها را به‌ترتیب می‌آموزد؛ در نقطه‌ای یک درخواست فراموشی برای یک وظیفه‌ی مشخص می‌رسد؛ سامانه باید رفتار مدل روی آن وظیفه را سرکوب کند و همان مدل باید روی وظیفه‌های باقی‌مانده در سطح پیشین خود کار کند. پس از حذف، دنباله ادامه می‌یابد و سامانه باید ضمن حفظ وظیفه‌های باقی‌مانده، توانایی یادگیری وظیفه‌های تازه را نیز حفظ کند.

[5] California Consumer Privacy Act (CCPA)
[6] Right to be Forgotten
[7] Machine Unlearning

۱-۲ تعریف مسئله

تنظیم مطالعه‌شده در این پایان‌نامه، یادگیری پیوسته‌ی افزایشی‌وظیفه[8] با درخواست‌های حذف در سطح وظیفه است. صورت‌بندی غیررسمی مسئله چنین است.

مدلی با پارامترهای θ دنباله‌ای از وظیفه‌ها $\mathcal{T} = \langle T_1, T_2, \ldots, T_n \rangle$ را به‌ترتیب می‌آموزد، به‌طوری‌که در هنگام آموزش وظیفه‌ی T_i فقط داده‌ی همان وظیفه به‌همراه یک حافظه‌ی بازپخش[9] با بودجه‌ی ثابت در اختیار است. در استنتاج، شناسه‌ی وظیفه در دسترس است و مدل از سرِ طبقه‌بندِ متناظر همان وظیفه استفاده می‌کند. در نقطه‌ای از این دنباله، یک درخواست فراموشی[10] برای وظیفه‌ی f می‌رسد. پس از پردازش این درخواست، دو نیاز هم‌زمان باید برآورده شود:

- **سرکوب هدف.** دقت مدل روی وظیفه‌ی f باید به سطح تصادفی آن وظیفه برسد؛ یعنی $A_f \to$ سطح تصادفی. برای یک وظیفه‌ی دوکلاسه این سطح ۰/۵ و برای یک وظیفه‌ی ده‌کلاسه ۰/۱ است.
- **حفظ نگه‌داری.** دقت مدل روی مجموعه‌ی وظایف فعال $\mathcal{T}_a = \mathcal{T} \setminus \{f\}$ باید تا حد امکان دست‌نخورده بماند؛ نه فقط به‌طور میانگین، بلکه برای بدترین وظیفه‌ی فعال نیز.

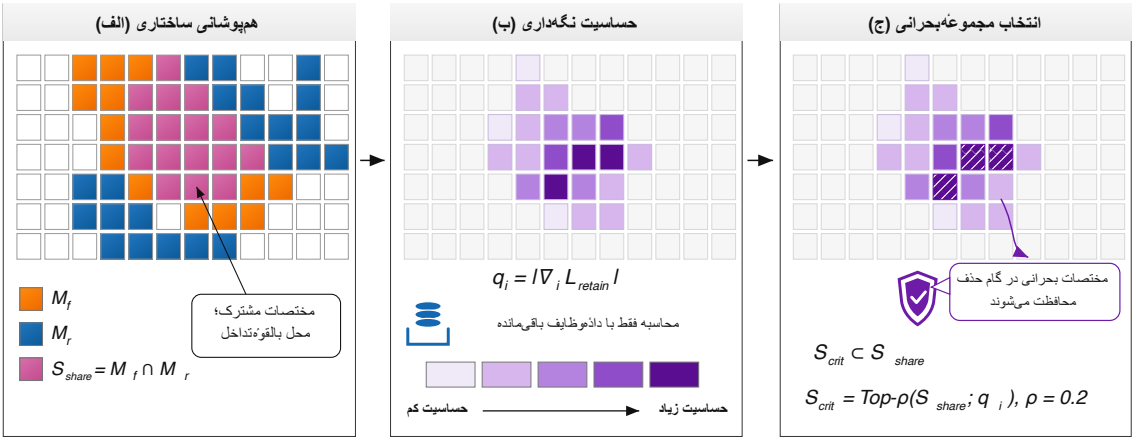
نکته‌ی مهم آن است که این پایان‌نامه سرکوب رفتار مدل روی یک وظیفه را با حذف داده هم‌ارز نمی‌گیرد. چیزی که در فرآیند درخواست حذف می‌شود، تنها مدخل بازپخش همان وظیفه در حافظه‌ی درون‌فرآیندی است؛ داده‌ی خام، نقاط‌بازرسی ذخیره‌شده و نسخه‌های بیرونی موضوع هیچ ادعایی در این کار نیستند. بخش ۱-۵ این حدود را به‌صورت صریح فهرست می‌کند.

۱-۳ چالش اصلی: هم‌پوشانی پارامتری

سختی مسئله از یک واقعیت ساختاری برمی‌خیزد. یک شبکه‌ی پرظرفیت، دانش هر وظیفه را روی تعداد زیادی وزن پخش می‌کند و این پخش‌شدگی میان وظایف مشترک است. اگر روی وظیفه‌ی هدف یک گام یادگیری‌زدایی بی‌قید اجرا شود — مثلا صعود گرادیانی[11] روی زیان همان وظیفه یا پاک‌سازی مبتنی بر فیشر[12] [۱۴] — همان گام، وزن‌هایی را هم جابه‌جا می‌کند که وظایف نگه‌داری‌شده به آن‌ها وابسته‌اند، و افت دقت آن‌ها یک اثر جانبی مستقیم است. این پدیده را در سراسر پایان‌نامه هم‌پوشانی پارامتری[13]

[8] Task-Incremental Continual Learning
[9] Replay Buffer
[10] Forgetting Request
[11] Gradient Ascent
[12] Fisher-based Scrubbing
[13] Parameter Overlap

می‌نامیم. شکل ۱-۲ شهود آن را نشان می‌دهد.



شکل ۱-۲: شهود هم‌پوشانی پارامتری. مسیرهای مربوط به وظیفه‌ی هدف فراموشی و وظایف نگه‌داری‌شده مجموعه‌ای از مختصه‌های وزن را به اشتراک می‌گذارند. هر به‌روزرسانی بی‌قید روی مسیر هدف، ناحیه‌ی مشترک را نیز جابه‌جا می‌کند و همین، منبع افت جانبی وظایف فعال است.

چارچوب یادگیری مادام‌العمر حریم‌خصوصی‌آگاه[۱۴] [۱۵] گام مهمی در برخورد با این مسئله برداشته است: دانش هر وظیفه را روی یک زیرشبکه‌ی پراکنده[۱۵] با ماسک دودویی ذخیره‌شده پیمانه‌ای[۱۶] می‌کند و در زمان درخواست، مختصه‌های همان زیرشبکه را بازنشانی یا بازسازی می‌کند و سپس وظایف نگه‌داری‌شده را ترمیم می‌کند. آنچه PALL انجام نمی‌دهد، تفکیک مختصه‌های مشترک بر حسب میزان حساسیت وظایف نگه‌داری‌شده به آن‌ها است. زیر هم‌پوشانی سنگین، همین تفکیک‌نکردن پرهزینه می‌شود: آشفتن یک ناحیه‌ی مشترکِ کوچک اما مهم می‌تواند چند وظیفه‌ی فعال را هم‌زمان آسیب بزند، در حالی که آشفتن یک ناحیه‌ی مشترکِ بزرگ اما کم‌اهمیت تقریبا بی‌هزینه است. مشاهده‌ی محرک این پایان‌نامه همین عدم‌تقارن است.

۱-۴ پرسش‌های پژوهش

پایان‌نامه حول پنج پرسش سازمان یافته است:

RQ1. آیا می‌توان با شناسایی صریح هم‌پوشانی ساختاری میان زیرشبکه‌ی وظیفه‌ی هدف و زیرشبکه‌های فعال، و محافظت از زیرمجموعه‌ی حساس آن، هم‌زمان به سرکوب هدف در سطح تصادفی و به نگه‌داری بهتر از PALL اصلی رسید؟

[14] Privacy-Aware Lifelong Learning (PALL)
[15] Sparse Subnetwork
[16] Modular

RQ2. در برابر مرجع جداسازی کامل — یعنی نگه‌داشتن یک شبکه‌ی مستقل برای هر وظیفه‌ی فعال، آن‌گونه که CLPU [۱۶] انجام می‌دهد — یک روش مدل‌مشترک چقدر می‌تواند نزدیک شود، و در ازای چه هزینه‌ای در حالت مقیم و زمان درخواست؟

RQ3. آیا ادعای انگیزشی روش، یعنی «محافظت با رشد هم‌پوشانی اهمیت بیشتری می‌یابد»، هم در محک توصیفی موقعیت درخواست و هم در جاروب پراکندگی زیرشبکه که دامنه‌ی گسترده‌ای از هم‌پوشانی اندازه‌گیری‌شده ایجاد می‌کند، پشتیبانی می‌شود؟

RQ4. کدام مؤلفه‌ی سازوکار پیشنهادی واقعا منشأ بهبود است؟ به‌طور مشخص، آیا فیلتر صریح اشتراک، رتبه‌بندی حساسیت، بودجه‌ی محافظت و جمله‌ی لنگر هر یک سهم قابل‌اندازه‌گیری دارند؟

RQ5. آیا انتقال سازوکار آگاه از هم‌پوشانی به یک مسیر کارآمد از نظر پارامتری — آداپترهای گلوگاهی روی بدنه‌ی منجمد — نقطه‌ی پایان درخواست را به‌طور قابل‌اندازه‌گیری بهبود می‌دهد، یا اثر مشاهده‌شده در آن مسیر با بازنشانی مسیر هدف توضیح داده می‌شود؟

پرسش پنجم آگاهانه به‌شکلی مطرح شده است که پاسخ منفی هم برای آن معتبر باشد. یکی از دستاوردهای این پایان‌نامه دقیقا همین است: نتیجه‌ی ممیزی مؤلفه‌ای برای مسیر آداپتر منفی است و به‌جای پنهان‌کردن، به‌عنوان یافته گزارش می‌شود.

۱-۵ حدود و دامنه‌ی ادعاها

برای پیش‌گیری از هرگونه بیش‌تفسیر، دامنه‌ی نتایج از ابتدا صریح می‌شود.

۱. **تجربی، نه گواهی‌شده.** هیچ نتیجه‌ای در این پایان‌نامه حذف گواهی‌شده[۱۷] به‌معنای [۱۷، ۱۸، ۱۹] را اثبات نمی‌کند. آن خط نظری برای مدل‌های محدب یا شرایط الگوریتمی خاص تضمین می‌دهد و به تنظیم غیرمحدب و دنباله‌ای این کار منتقل نمی‌شود.

۲. **سطح وظیفه، نه سطح رکورد.** واحد فراموشی، یک وظیفه با شناسه‌ی مشخص است. حذف یک نمونه‌ی تک یا یک کلاس درون وظیفه در دامنه‌ی این کار نیست.

۳. **در فرآیند، نه در ذخیره‌سازی.** تنها مدخل بازپخش وظیفه‌ی هدف از حافظه‌ی درون‌فرآیندی حذف می‌شود. هیچ ادعایی در مورد پاک‌شدن داده‌ی خام روی دیسک، نقاط‌بازرسی، پشتیبان‌ها یا نسخه‌های بیرونی مطرح نیست.

[17] Certified Removal

۴. **شناسه‌ی وظیفه در استنتاج لازم است.** رابط روش به شناسه‌ی وظیفه نیاز دارد و سناریوی افزایشی‌کلاس[18] بدون شناسه پوشش داده نمی‌شود.

۵. **حمله‌ی عضویت کم‌توان.** ممیزی حریم خصوصی این کار بر حمله‌ی استنتاج عضویت[19] مبتنی بر آستانه‌ی زیان استوار است. نتیجه‌ی نزدیک به ۰/۵ در چنین آزمونی یک نتیجه‌ی تهیِ کم‌توان است و شاهد حذف نیست؛ حمله‌های قوی‌تر [۱۱] یا ممیزی صوری [۲۰] لازم است.

۶. **زمان نرمال‌نشده.** زمان درخواست روی یک پیکربندی سخت‌افزاری مشخص اندازه‌گیری شده و نسبت به سخت‌افزار نرمال نشده است؛ بنابراین برای مقایسه‌ی مرتبه‌ای مفید است، نه برای مقایسه‌ی دقیق کارایی.

۷. **نمونه‌ی کوچک.** بخشی از تحلیل‌ها با دو یا سه بذر انجام شده است. این موارد به‌صراحت توصیفی برچسب می‌خورند و تنها جایی ادعای معناداری مطرح می‌شود که آزمون دقیق جفتی گزارش شده باشد.

۱-۶ مشارکت‌های پژوهش

مشارکت‌های این پایان‌نامه در چهار محور خلاصه می‌شود.

یک. محافظت از وزن‌های مشترکِ حساس به نگه‌داری. روش EPALL معرفی می‌شود که هم‌پوشانی ساختاری میان زیرشبکه‌ی فراموش‌شونده و اجتماع زیرشبکه‌های فعال را می‌سنجد، مختصه‌های مشترک را با بزرگی گرادیان وظایف نگه‌داری‌شده رتبه‌بندی می‌کند و زیرمجموعه‌ی بحرانی را در طول ترمیم با یک جمله‌ی لنگر ℓ_2 مهار می‌کند. فصل ۳ این سازوکار را به‌طور کامل توصیف می‌کند.

دو. ارزیابی هم‌تراز و دو محک مکمل برای هم‌پوشانی. یک مقایسه میان یازده روش و روی هشت بذر روی Split-CIFAR انجام می‌شود که در آن همه‌ی روش‌ها درون هر بذر، دنباله‌ی درخواست یکسانی می‌بینند و چهار معیار نگه‌داری، فراموشی، سرکوب هدف و زمان درخواست به‌طور هم‌زمان و با تحلیل جفتی گزارش می‌شوند. علاوه بر آن، یک محک ۳۰۰ اجرایی با پنج درجه‌ی از پیش مشخص‌شده‌ی موقعیت درخواست و یک جاروب پراکندگی با هویت و موقعیت ثابت وظیفه‌ی هدف گزارش می‌شود. محک نخست تنها شاخصی جانشین برای هم‌پوشانی مورد انتظار است؛ جاروب دوم دامنه‌ی گسترده‌ای

[18] Class-Incremental
[19] Membership Inference Attack

از هم‌پوشانی ساختاری اندازه‌گیری‌شده ایجاد می‌کند، اما تغییر هم‌زمان ظرفیت اجازه‌ی تفسیر علّی خالص نمی‌دهد.

سه. ممیزی سازوکار با نتیجه‌ی منفی صریح. برای هر دو مسیر پیشنهادی، کنترل‌های هم‌تراز طراحی می‌شود که هر بار فقط یک مؤلفه را تغییر می‌دهند. برای EPALL این کنترل‌ها نشان می‌دهند رتبه‌بندی حساسیت به‌روشنی سود می‌رساند، در حالی که دو جزء دیگر سهم مستقل قابل‌اندازه‌گیری نشان نمی‌دهند. برای PALL-Adapter، کنترل‌ها سرکوب هدف را به بازنشانی مسیر هدف نسبت می‌دهند و نه به ماسک نرم؛ این یافته‌ی منفی به‌عنوان بخشی از نتایج ارائه می‌شود.

چهار. حسابداری صریح هزینه‌ها. به‌جای تکیه بر نسبت پارامترهای به‌روزشده به‌عنوان شاخص کارایی، حالت مقیم تنسوری به‌طور مستقیم و روی دنباله‌های هم‌تراز حسابداری می‌شود و تفاوت آن با نسبت دامنه‌ی ترمیم به‌صراحت توضیح داده می‌شود.

۱–۷ معیارهای موفقیت

معیارهای ارزیابی این پژوهش در فصل ۴ بر مبنای موارد زیر گزارش و تفسیر می‌شوند:

- دقت وظیفه‌ی فراموش‌شده باید در فاصله‌ی کوچکی از سطح تصادفی همان وظیفه قرار گیرد؛ معیار خوانش، قدرمطلق فاصله تا سطح تصادفی است و نه مقدار خام دقت.
- دقت نگه‌داری باید نسبت به PALL اصلی، در تحلیل جفتی روی بذرهای یکسان بهبود یابد.
- بیشترین افت میان وظایف فعال باید نسبت به PALL اصلی کاهش یابد؛ این معیار عمدا علامت‌دار و بی‌بریدگی گزارش می‌شود.
- هزینه‌ی زمان درخواست باید گزارش شود، حتی جایی که به زیان روش پیشنهادی است.

۱–۸ ساختار پایان‌نامه

بقیه‌ی پایان‌نامه چنین سازمان یافته است. فصل ۲ مفاهیم پایه و کارهای پیشین را در چهار خط — یادگیری پیوسته، یادگیری‌زدایی ماشینی، یادگیری کارآمد پارامتری و ارزیابی حریم خصوصی — مرور می‌کند و

در پایان، جایگاه این کار را در قالب یک جدول مقایسه‌ی مفهومی و یک بیان صریح از خلأ پژوهشی مشخص می‌سازد. فصل ۳ روش پیشنهادی را ارائه می‌کند: صورت‌بندی رسمی، سازوکار EPALL با الگوریتم و تحلیل پیچیدگی، مسیر اکتشافی PALL-Adapter به‌همراه یک تحلیل نظری با دامنه‌ی دقیقا مشخص‌شده، و در انتها توضیح این‌که چرا آن تحلیل، افت انتهابه‌انتها را کران نمی‌زند. فصل ۴ طراحی آزمایش، معیارها و همه‌ی نتایج را گزارش می‌کند، از مقایسه‌ی اصلی و آزمون‌های معناداری جفتی تا محک موقعیت درخواست، حسابداری حافظه، ممیزی حریم خصوصی و مطالعات حساسیت. فصل ۵ یافته‌ها را در قالب پاسخ به پرسش‌های پژوهش جمع می‌بندد، محدودیت‌ها را فهرست می‌کند و مسیرهای آینده را پیشنهاد می‌دهد. پیوست ۵–۶ جزئیات بازتولیدپذیری، پیکربندی کامل ابرپارامترها و نگاشت گروه‌های آزمایش به برچسب‌های مستقل را در بر می‌گیرد.

فصل ۲

مفاهیم پایه و مرور کارهای پیشین

این فصل زمینه‌ی لازم برای فهم روش پیشنهادی را فراهم می‌کند. ساختار آن از عام به خاص است: نخست مفاهیم پایه‌ی یادگیری عمیق و اجزایی که در آزمایش‌ها به‌کار می‌روند مرور می‌شوند؛ سپس یادگیری پیوسته و خانواده‌های راه‌حل آن؛ آنگاه یادگیری‌زدایی ماشینی و تلاقی آن با یادگیری پیوسته؛ پس از آن یادگیری کارآمد پارامتری؛ و در پایان روش‌شناسی ارزیابی فراموشی و حریم خصوصی. فصل با یک جدول مقایسه‌ی مفهومی و بیان صریح خلأ پژوهشی بسته می‌شود. برای مرورهای جامع یادگیری پیوسته می‌توان به [۴، ۵، ۲۱، ۲۲] و برای مرورهای یادگیری‌زدایی به [۲۳، ۲۴] رجوع کرد.

۲-۱ مفاهیم پایه

۲-۱-۱ یادگیری بانظارت و بهینه‌سازی گرادیانی

یک شبکه‌ی طبقه‌بند را با $f(\cdot\,;\theta)$ نشان می‌دهیم که ورودی x را به برداری از لاجیت‌ها[1] می‌نگارد. آموزش بانظارت، کمینه‌سازی مخاطره‌ی تجربی[2] است:

$$\widehat{R}(\theta) = \frac{1}{|D|} \sum_{(x,y)\in D} \ell\big(f(x;\theta), y\big), \tag{۲-۱}$$

که در آن ℓ معمولا زیان آنتروپی متقاطع[3] است. کمینه‌سازی با گرادیان کاهشی تصادفی[4] [۲۵] و به‌روزرسانی $\theta \leftarrow \theta - \eta \nabla_\theta \widehat{R}_B(\theta)$ روی دسته‌های کوچک B انجام می‌شود. همه‌ی آزمایش‌های این

[1] Logits
[2] Empirical Risk
[3] Cross-Entropy
[4] Stochastic Gradient Descent (SGD)

پایان‌نامه از SGD با تکانه استفاده می‌کنند؛ دلیل این انتخاب، حذف اثر تطبیقی بهینه‌سازهای مبتنی بر ممان دوم است، چون در سناریوی فراموشی، حالت انباشته‌ی بهینه‌ساز خودش می‌تواند اثری از وظیفه‌ی حذف‌شده حمل کند.

بردار گرادیان در این پایان‌نامه دو نقش دارد و تفکیک این دو نقش مهم است. نقش اول، نقش متعارف آن به‌عنوان جهت به‌روزرسانی است. نقش دوم — که سازوکار پیشنهادی بر آن استوار است — استفاده از بزرگی مؤلفه‌های گرادیان به‌عنوان یک سنجه‌ی محلی از اهمیت پارامتر[۵] است: اگر $|\partial_i \mathcal{L}_{\text{retain}}|$ بزرگ باشد، جابه‌جایی کوچک در مختصه‌ی i زیان وظایف نگه‌داری‌شده را به‌طور محسوس تغییر می‌دهد.

۲-۱-۲ سنجه‌های اهمیت پارامتر

ادبیات یادگیری پیوسته چند سنجه‌ی اهمیت را پیشنهاد کرده است و انتخاب میان آن‌ها یک مصالحه‌ی دقت–هزینه است. ماتریس اطلاعات فیشر[۶] در EWC [۷] از میانگین مربع گرادیان لگ‌درست‌نمایی برآورد می‌شود و مبنای نظری روشنی دارد، اما محاسبه‌ی آن یک گذر کامل روی داده لازم دارد. هوشمندی سیناپسی[۷] [۲۶] اهمیت را در طول مسیر آموزش انباشته می‌کند و سیناپس‌های آگاه از حافظه[۸] [۲۷] از حساسیت خروجی به آشفتگی وزن استفاده می‌کنند. توابع نفوذ[۹] [۲۸] اثر یک نمونه‌ی آموزشی بر پارامترها را تقریب می‌زنند و پیوند مفهومی نزدیکی با یادگیری‌زدایی دارند.

در این پایان‌نامه، سنجه‌ی پیش‌فرض بزرگی گرادیان روی حافظه‌ی بازپخش وظایف نگه‌داری‌شده است. دلیل این انتخاب صرفا عمل‌گرایانه است: یک گذر پس‌رو روی حافظه‌ی محدود کافی است و هزینه‌ی آن نسبت به حلقه‌ی ترمیم ناچیز است. در بخش ۲-۱۲-۴ یک جانشین مبتنی بر انرژی تعارض گرادیانی[۱۰] نیز به‌طور تجربی سنجیده می‌شود که نتیجه‌ی آن وابسته به مجموعه‌داده است.

۲-۱-۳ معماری‌های استفاده‌شده

آزمایش‌ها بر دو خانواده‌ی معماری استوارند. خانواده‌ی نخست شبکه‌های باقی‌مانده[۱۱] یا ResNet [۲۹] است. اتصال باقی‌مانده‌ی $y = \mathcal{F}(x) + x$ مسیری کوتاه برای جریان گرادیان ایجاد می‌کند و آموزش شبکه‌های عمیق را پایدار می‌سازد. در این کار ResNet-18 برای CIFAR-10 و TinyImageNet و

[5] Parameter Importance
[6] Fisher Information Matrix
[7] Synaptic Intelligence
[8] Memory Aware Synapses
[9] Influence Functions
[10] Gradient-Conflict Energy
[11] Residual Networks

اجراهای پیش‌آموزش‌دیده، و ResNet-34 برای اجراهای اصلی CIFAR-100 به‌کار می‌رود. خانواده‌ی دوم ترنسفورمر بینایی[۱۲] [۳۰] است که تصویر را به وصله‌ها تقسیم و با لایه‌های توجه پردازش می‌کند؛ از این خانواده یک نسخه‌ی کوچک به‌عنوان آزمون انتقال میان‌معماری استفاده می‌شود.

مقدارگذاری اولیه نیز در جای‌جای روش نقش دارد. بازنشانی مختصه‌های وظیفه‌ی هدف از توزیع کایمینگ[۱۳] [۳۱] نمونه می‌گیرد؛ همین انتخاب در آداپترها هم برای تصویر پایین‌رو تکرار می‌شود، در حالی که تصویر بالارو با صفر مقدار می‌گیرد تا آداپتر در لحظه‌ی تولد، تابع همانی باشد.

۲-۱-۴ حافظه‌ی محدود و بازپخش

فرض دسترسی‌نداشتن به داده‌ی وظایف گذشته در عمل بسیار سخت‌گیرانه است. راه‌حل رایج، نگه‌داشتن یک حافظه‌ی اپیزودیک[۱۴] کوچک است. در این کار بودجه‌ی بازپخش ۵۰۰ نمونه است و — نکته‌ی مهم — این بودجه از پیش میان وظایف افراز می‌شود و پس از یک درخواست حذف، سهم آزادشده به وظایف دیگر بازتوزیع نمی‌شود. دلیل این محدودیت، جلوگیری از یک اثر مخدوش‌کننده است: اگر بودجه بازتوزیع شود، بخشی از بهبود مشاهده‌شده پس از فراموشی می‌تواند صرفا ناشی از بزرگ‌ترشدن حافظه‌ی وظایف باقی‌مانده باشد و نه ناشی از سازوکار فراموشی.

۲-۲ یادگیری پیوسته

۲-۲-۱ سه سناریو

پیکربندی‌های یادگیری پیوسته را می‌توان بر اساس اطلاعات موجود در استنتاج به سه سناریو تقسیم کرد [۲۱]: افزایشی‌وظیفه[۱۵] که شناسه‌ی وظیفه در استنتاج در دسترس است؛ افزایشی‌دامنه[۱۶] که مجموعه‌ی برچسب‌ها ثابت است و توزیع ورودی تغییر می‌کند؛ و افزایشی‌کلاس[۱۷] که مدل باید بدون شناسه، میان همه‌ی کلاس‌های دیده‌شده تمایز بگذارد. این پایان‌نامه در سناریوی نخست کار می‌کند. این انتخاب یک محدودیت واقعی است، اما برای مسئله‌ی حذف در سطح وظیفه ضروری است: مفهوم «فراموش‌کردن وظیفه‌ی f» تنها زمانی خوش‌تعریف است که مرزهای وظیفه در سطح رابط سامانه شناخته باشند.

[12] Vision Transformer
[13] Kaiming Initialization
[14] Episodic Memory
[15] Task-Incremental
[16] Domain-Incremental
[17] Class-Incremental

تفاوت این سه سناریو را می‌توان دقیق‌تر بیان کرد. فرض کنید مدل تابعی از ورودی x و شناسه‌ی وظیفه t باشد. در سناریوی افزایشی‌وظیفه، مدل $p(y \mid x, t)$ را می‌آموزد و در استنتاج هر دو آرگومان در دسترس‌اند؛ عملا این یعنی هر وظیفه می‌تواند سر طبقه‌بند خودش را داشته باشد و مسئله‌ی تمایز میان کلاس‌های وظایف مختلف اصلا مطرح نمی‌شود. در سناریوی افزایشی‌دامنه، فضای برچسب ثابت است و مدل همچنان $p(y \mid x)$ را می‌آموزد، اما توزیع ورودی در طول زمان جابه‌جا می‌شود. دشوارترین حالت، سناریوی افزایشی‌کلاس است: مدل باید $p(y \mid x)$ را روی *اجتماع* همه‌ی کلاس‌های دیده‌شده بیاموزد بدون آنکه بداند نمونه از کدام وظیفه آمده است، و بنابراین باید مرزهایی میان کلاس‌هایی بکشد که هرگز در یک دسته‌ی آموزشی کنار هم ظاهر نشده‌اند.

این تفکیک برای مسئله‌ی حذف پیامد مستقیم دارد. واحد حذف در این پایان‌نامه یک وظیفه‌ی کامل است، و «وظیفه» تنها در سناریوی نخست یک شیء صریح در رابط سامانه است. در سناریوی افزایشی‌کلاس، درخواست حذف باید در سطح کلاس یا رکورد بیان شود، که مسئله‌ی متفاوتی است: در آنجا نمی‌توان یک سر طبقه‌بند یا یک زیرشبکه‌ی مشخص را کنار گذاشت، چون دانش کلاس حذف‌شونده در همان مرزهایی پخش است که کلاس‌های باقی‌مانده را از هم جدا می‌کنند. بنابراین انتخاب سناریوی افزایشی‌وظیفه در این کار یک ساده‌سازی دلبخواهی نیست، بلکه شرط خوش‌تعریفی صورت‌مسئله است؛ در عوض، تعمیم روش به دو سناریوی دیگر در بخش ۵-۴ به‌عنوان محدودیت ثبت می‌شود.

۲-۲-۲ روش‌های مبتنی بر تنظیم

نخستین خانواده‌ی راه‌حل، جابه‌جایی پارامترهای مهم را جریمه می‌کند. EWC [۷] جمله‌ی $\sum_i F_i(\theta_i - \theta_i^*)^2$ را با وزن‌های فیشر اضافه می‌کند؛ SI [۲۶] و MAS [۲۷] همان قالب را با سنجه‌های اهمیت متفاوت پیاده می‌کنند. یادگیری بدون فراموشی[18] یا LwF [۳۲] به‌جای جریمه‌ی وزن، خروجی مدل جدید را به خروجی مدل قدیم نزدیک نگه می‌دارد و در واقع یک تقطیر دانش[19] [۳۳] خودارجاع است.

شهود مشترک این خانواده ساده است: اگر بدانیم کدام پارامترها برای وظایف گذشته حیاتی‌اند، می‌توانیم به‌جای نگه‌داشتن داده، خودِ پارامترها را نگه داریم. اختلاف روش‌ها در پاسخ به این پرسش است که «اهمیت» چگونه اندازه‌گیری شود. EWC از قطر ماتریس اطلاعات فیشر استفاده می‌کند، که برابر امید مربع گرادیان لگاریتم درست‌نمایی است و بنابراین حساسیت *محلی* خروجی مدل به هر پارامتر را می‌سنجد. SI همان کمیت را به‌صورت برخط و در طول مسیر بهینه‌سازی انباشته می‌کند و به یک گذر جداگانه نیاز ندارد. LwF مسیر متفاوتی می‌رود و به‌جای پارامترها، *رفتار* مدل را مقید می‌کند: خروجی نرم

[18] Learning without Forgetting
[19] Knowledge Distillation

مدل پیشین روی داده‌ی جدید به‌عنوان هدف تقطیر به‌کار می‌رود، پس هیچ داده‌ای از وظایف گذشته لازم نیست.

دو محدودیت مشترک این خانواده در ادبیات ثبت شده است. نخست، جریمه سراسری است و روی همه‌ی پارامترها اعمال می‌شود؛ با افزایش تعداد وظایف، جمله‌های جریمه روی هم انباشته می‌شوند و ظرفیت یادگیری وظیفه‌ی تازه را می‌بلعند. دوم، تقریب قطری فیشر همبستگی میان پارامترها را نادیده می‌گیرد، در حالی که آسیب واقعی اغلب از حرکت *هم‌زمان* گروهی از وزن‌ها می‌آید.

پیوند این خانواده با کار حاضر نزدیک است، اما جهت آن معکوس. جمله‌ی لنگر ℓ_2 در روش پیشنهادی از نظر شکل ریاضی همان قالب EWC است؛ تفاوت در این است که در EWC هدف، محافظت در برابر *یادگیری* وظیفه‌ی بعدی است و مجموعه‌ی محافظت‌شده کل پارامترهای مهم است، در حالی که در این کار هدف، محافظت در برابر *فراموش‌کردن* است و مجموعه‌ی محافظت‌شده به هم‌پوشانی ساختاری میان زیرشبکه‌ی هدف و زیرشبکه‌های فعال محدود می‌شود. همچنین یک نکته‌ی تجربی مهم در فصل ۴ روشن می‌شود: EWC و LwF در معیار بیشترین افت عالی‌اند اما در سرکوب وظیفه‌ی هدف شکست می‌خورند، چون سازوکار آن‌ها اساسا برای حفظ ساخته شده است، نه برای حذف.

۲-۲-۳ روش‌های مبتنی بر بازپخش

خانواده‌ی دوم، نمونه‌های گذشته را در حافظه نگه می‌دارد و در آموزش وظیفه‌ی جدید بازپخش می‌کند. بازپخش تجربه[۲۰] ساده‌ترین شکل آن است و iCaRL [۳۴] آن را با نمونه‌های شاخص و طبقه‌بندی نزدیک‌ترین میانگین ترکیب می‌کند. بازپخش تجربه‌ی تاریک[۲۱] یا DER++ [۳۵] علاوه بر برچسب‌ها، لاجیت‌های ذخیره‌شده را نیز تطبیق می‌دهد و به همین دلیل سیگنال غنی‌تری از رفتار گذشته‌ی مدل حمل می‌کند. جایگزین مولد این خانواده، بازپخش نمونه‌های سنتزی است [۳۶].

زیان ترمیمی روش پیشنهادی مستقیما از این خط می‌آید: جمله‌ی $\mathcal{L}_{DER}$ همان تطبیق لاجیت‌های ذخیره‌شده است. دلیل این انتخاب آن است که پس از یک درخواست حذف، هدف ما بازگرداندن *رفتار* پیشین مدل روی وظایف فعال است و لاجیت‌های ذخیره‌شده دقیقا همان رفتار را ثبت کرده‌اند.

اندازه‌ی حافظه در این خانواده یک پارامتر طراحی تعیین‌کننده است و در تنظیم حذف، معنای تازه‌ای پیدا می‌کند: حافظه‌ی بازپخش خودش نسخه‌ای از داده‌ی کاربر است. اگر درخواست حذف یک وظیفه برسد اما مدخل بازپخش آن باقی بماند، داده‌ی حذف‌شونده همچنان در سامانه حاضر است و هر ادعای حذفی بی‌معنا می‌شود. به همین دلیل در این پایان‌نامه حذف مدخل بازپخش وظیفه‌ی هدف بخشی از تعریف عملیات

[20] Experience Replay
[21] Dark Experience Replay

حذف است و نه یک جزئیات پیاده‌سازی. در مقابل، بودجه‌ی حافظه پس از حذف *بازتوزیع نمی‌شود*؛ این انتخاب عمدی است تا مقایسه‌ی روش‌ها زیر بودجه‌ی یکسان باقی بماند.

۲-۲-۴ تصویر گرادیان

حافظه‌ی اپیزودیک گرادیانی[۲۲] یا GEM [۳۷] و نسخه‌ی میانگین‌گیری‌شده‌ی آن A-GEM [۳۸] به‌روزرسانی وظیفه‌ی جدید را طوری تصویر می‌کنند که ضرب داخلی آن با گرادیان وظایف گذشته منفی نشود. این خانواده از نظر روحیه نزدیک‌ترین کار به سازوکار این پایان‌نامه است: در هر دو، گرادیان وظایف نگه‌داری‌شده به‌روزرسانی را مقید می‌کند. تفاوت کلیدی در محل اعمال قید است. GEM یک تصویر *سراسری* در فضای گرادیان انجام می‌دهد و همه‌ی مختصه‌ها را به‌طور یکنواخت تحت تأثیر قرار می‌دهد؛ روش این پایان‌نامه یک ناحیه‌ی *انتخاب‌شده* از مختصه‌های مشترک را شناسایی و در همان‌جا مهار می‌کند. این تفاوت هم مفهومی است و هم عملی: انتخاب ناحیه‌ای اجازه می‌دهد بقیه‌ی به‌روزرسانی آزادانه در خدمت فراموشی باقی بماند.

۲-۲-۵ جداسازی پارامتر و زیرشبکه‌های پراکنده

خانواده‌ی چهارم، پارامترها را به‌ازای هر وظیفه جدا می‌کند. PackNet [۳۹] پس از آموزش هر وظیفه هرس می‌کند و وزن‌های باقی‌مانده را قفل می‌کند؛ ماسک توجه سخت[۲۳] یا HAT [۴۰] ماسک‌های واحدمحور یاد می‌گیرد؛ و SupSup [۴۱] تنها ماسک‌های دودویی را روی یک بدنه‌ی تصادفی و ثابت می‌آموزد. شبکه‌های پیش‌رونده[۲۴] [۴۲] در انتهای دیگر این طیف قرار دارند و ظرفیت را به‌ازای هر وظیفه رشد می‌دهند.

مبنای نظری این خانواده به فرضیه‌ی بلیت بخت‌آزمایی[۲۵] [۴۳] و به نتیجه‌ی مکمل آن [۴۴] می‌رسد: زیرشبکه‌های قابل‌آموزش با کارایی خوب، درون شبکه‌های چگال و حتی درون شبکه‌های تصادفی وجود دارند. یادگیری پیوسته‌ی زیرشبکه‌ی برنده[۲۶] یا WSN [۴۵] این ایده را به یادگیری پیوسته می‌آورد و برای هر وظیفه یک ماسک دودویی می‌آموزد. PALL ماسک‌های هر ـ وظیفه‌ی خود را از همین خط به ارث می‌برد و همین ارث‌بری است که مفهوم *هم‌پوشانی ساختاری* میان دو وظیفه را خوش‌تعریف می‌کند: هم‌پوشانی، اشتراک مجموعه‌ای دو ماسک است و لازم نیست با یک سنجه‌ی پیوسته تقریب زده شود. این نکته پایه‌ی مستقیم بخش ۳-۳ است.

[22] Gradient Episodic Memory
[23] Hard Attention to the Task
[24] Progressive Networks
[25] Lottery Ticket Hypothesis
[26] Winning Subnetwork Continual Learning

چارچوب PALL [۱۵] که این پایان‌نامه بر آن سوار می‌شود، همین ایده را برای حذف به‌کار می‌گیرد. به‌ازای هر وظیفه یک ماسک دودویی پراکنده روی وزن‌های شبکه آموخته و ذخیره می‌شود، به‌گونه‌ای که وظیفه تنها از راه زیرشبکه‌ی خودش محاسبه شود. درخواست حذف آن‌گاه یک عمل *ساختاری* است و نه یک بهینه‌سازی: ماسک وظیفه‌ی هدف بازنشانی می‌شود، سر طبقه‌بند متناظر پاک می‌گردد، و مدخل بازپخش آن وظیفه از حافظه حذف می‌شود. مزیت این طراحی آن است که هزینه‌ی حذف مستقل از اندازه‌ی داده است و برخلاف روش‌های مبتنی بر صعود گرادیان، به گام بهینه‌سازی روی داده‌ی حذف‌شونده نیاز ندارد.

نقطه‌ی ضعف همین‌جا آشکار می‌شود و انگیزه‌ی این پایان‌نامه است. ماسک‌های وظایف مختلف *مجزا نیستند*؛ در یک شبکه‌ی با پراکندگی ثابت، هرچه تعداد وظایف بیشتر شود اشتراک ماسک‌ها ناگزیر بزرگ‌تر می‌شود. بازنشانی زیرشبکه‌ی هدف بنابراین مختصاتی را نیز لمس می‌کند که وظایف فعال هنوز به آن‌ها تکیه دارند، و همین منشأ آسیب جانبی است. PALL برای جبران، پس از بازنشانی یک گام ترمیم روی داده‌ی بازپخش وظایف باقی‌مانده اجرا می‌کند؛ اما این ترمیم هیچ تمایزی میان مختصات مشترک و غیرمشترک نمی‌گذارد و هیچ سازوکاری برای *پیش‌گیری* از آسیب ندارد. کار حاضر دقیقا همین شکاف را هدف می‌گیرد.

۲-۳ یادگیری‌زدایی ماشینی

۲-۳-۱ فراموشی دقیق و بازآموزی

مرجع مفهومی یادگیری‌زدایی، بازآموزی کامل مدل روی داده‌ی باقی‌مانده است. چون این کار پرهزینه است، SISA [۱۳] داده را به بخش‌های جدا افراز می‌کند، یک عضو گروه را روی هر بخش آموزش می‌دهد و در زمان حذف تنها عضو آسیب‌دیده را بازآموزی می‌کند. آموزش فراموش‌کار[۲۷] [۴۶] به‌روزرسانی‌های هر دسته را ذخیره می‌کند تا بعدا کم شود. کارهای دیگری برای مدل‌های ساده مانند خوشه‌بندی k-means حذف دقیق ارائه کرده‌اند [۴۷].

معنای دقیق «دقیق» در این خانواده اهمیت دارد. یک سازوکار حذف را دقیق می‌نامیم هرگاه توزیع مدل خروجی پس از حذف نمونه‌ی z، با توزیع مدلی که از ابتدا بدون z آموزش دیده است *یکسان* باشد. این تعریف بر توزیع مدل‌ها بنا شده و نه بر رفتار یک مدل منفرد؛ به همین دلیل صرفِ افت دقت روی داده‌ی حذف‌شده، شاهدی برای حذف دقیق نیست. بازآموزی کامل به‌طور بدیهی این تعریف را برآورده می‌کند و

[27] Amnesiac Training

به همین دلیل مرجع مفهومی این حوزه است.

هزینه‌ی این مرجع همان چیزی است که کل ادبیات را شکل داده است. بازآموزی کامل به نگه‌داشتن دائمی داده‌ی خام و پرداخت دوباره‌ی تمام هزینه‌ی آموزش در هر درخواست نیاز دارد، و در تنظیم پیوسته که درخواست‌ها به‌صورت جریانی می‌رسند این هزینه به‌ازای هر درخواست تکرار می‌شود. رویکردهای افرازی مانند SISA هزینه را کاهش می‌دهند اما آن را با دو چیز معاوضه می‌کنند: کاهش دقت، چون هر عضو گروه تنها بخشی از داده را می‌بیند، و رشد حالت مقیم، چون باید چند مدل هم‌زمان نگه‌داری شوند. همین معاوضه است که در این پایان‌نامه نیز، در قالب مقایسه با مرجع جداسازی، دوباره ظاهر می‌شود.

۲-۳-۲ فراموشی تقریبی

خانواده‌ی تقریبی به‌جای بازآموزی، وزن‌ها را مستقیما ویرایش می‌کند. [۱۴] یک به‌روزرسانی مبتنی بر فیشر برای پاک‌کردن اطلاعات پیشنهاد می‌دهد؛ [۴۸] روش‌های حذف تقریبی برای مدل‌های خطی ارائه می‌کند؛ فراموشی مرزی[۲۸] [۴۹] مرزهای تصمیم خروجی را جابه‌جا می‌کند تا سرعت به‌دست آورد؛ و [۵۰] از یک جفت معلم خوب و بد برای هدایت فراموشی استفاده می‌کند. دو کار اخیرتر به این پایان‌نامه نزدیک‌ترند و در آزمایش‌ها به‌عنوان خط پایه به‌کار می‌روند: یادگیری‌زدایی برجستگی[۲۹] یا SalUn [۵۱] به‌روزرسانی را به وزن‌های پرنفوذ محدود می‌کند، و میراکردن انتخابی سیناپسی[۳۰] یا SSD [۵۲] پارامترها را بر پایه‌ی اهمیت فیشری میرا می‌کند و برای این کار به هیچ گام آموزشی نیاز ندارد.

SalUn از منظر این پایان‌نامه اهمیت مفهومی خاصی دارد، چون آن هم به‌روزرسانی را روی یک زیرمجموعه‌ی انتخاب‌شده متمرکز می‌کند. اما جهت انتخاب متفاوت است: SalUn مختصه‌های پرنفوذ را برای *تغییردادن* انتخاب می‌کند، در حالی که روش این پایان‌نامه مختصه‌های حساس به نگه‌داری را برای *تغییرندادن* انتخاب می‌کند.

۲-۳-۳ تضمین‌های صوری و مرزهای آن

یک خط نظری موازی برای مدل‌های محدب تضمین حذف ارائه می‌دهد. [۱۷] مفهوم حذف داده‌ی گواهی‌شده[۳۱] را با تعریفی شبیه به حریم خصوصی تفاضلی[۳۲] [۵۳] معرفی می‌کند؛ [۱۸] الگوریتم‌های مبتنی بر کاهش گرادیان با تضمین ارائه می‌دهد و [۱۹] مصالحه‌ی میان تعمیم و حذف را بررسی می‌کند.

[28] Boundary Unlearning
[29] Saliency Unlearning
[30] Selective Synaptic Dampening
[31] Certified Data Removal
[32] Differential Privacy

باید صریح گفت که این تضمین‌ها به تنظیم این پایان‌نامه منتقل نمی‌شوند. آموزش دنباله‌ای یک شبکه‌ی عمیق غیرمحدب، با ماسک‌های وابسته به تاریخ و یک حافظه‌ی بازپخش که خودش تابعی از دنباله‌ی درخواست‌ها است، هیچ‌یک از پیش‌فرض‌های آن قضایا را برآورده نمی‌کند. به همین دلیل، همه‌ی ادعاهای این پایان‌نامه تجربی برچسب می‌خورند. بحث سازوکارهای «یادگیری برای فراموش‌کردن» در [۵۴] آمده است.

برای آنکه روشن باشد چه چیزی منتقل نمی‌شود، صورت آرام‌شده‌ی این تضمین را می‌آوریم. الگوریتم حذف U را نسبت به الگوریتم آموزش A و مجموعه‌داده‌ی D یک حذف (ϵ, δ) می‌نامیم هرگاه برای هر نمونه‌ی $z \in D$ و هر مجموعه‌ی اندازه‌پذیر S از مدل‌ها، رابطه‌ی

$$\Pr\left[U(A(D), z) \in S\right] \leqslant e^{\epsilon} \Pr\left[A(D \setminus z) \in S\right] + \delta \qquad (۲-۲)$$

و رابطه‌ی متقارن آن برقرار باشد. این صورت‌بندی عمدا به تعریف حریم خصوصی تفاضلی [۵۳] شبیه است و مانند آن، گزاره‌ای درباره‌ی *توزیع* خروجی الگوریتم است و نه درباره‌ی یک اجرای منفرد.

سه پیش‌فرض این قضایا در تنظیم ما نقض می‌شوند. نخست، اثبات‌ها به محدب‌بودن تابع هزینه و یکتایی کمینه تکیه دارند تا بتوانند فاصله‌ی پارامترها را کران‌دار کنند؛ یک شبکه‌ی عمیق هیچ‌کدام را ندارد. دوم، فرض می‌شود مسیر آموزش تنها از راه داده به آن وابسته است، در حالی که اینجا ماسک‌های زیرشبکه خودشان تابعی از ترتیب وظایف‌اند و تاریخ اجرا را در ساختار مدل رمز می‌کنند. سوم، حافظه‌ی بازپخش محدود، خود تابعی از دنباله‌ی درخواست‌هاست؛ پس حتی اگر به‌روزرسانی حذف کران‌دار می‌بود، وضعیت اولیه‌ای که روی آن اعمال می‌شود به داده‌ی حذف‌شونده وابسته است. به همین سه دلیل، در این پایان‌نامه هیچ عدد ϵ گزارش نمی‌شود و هر ادعای حذف، تجربی و مشروط بیان می‌گردد.

۲-۳-۴ یادگیری‌زدایی در بستر یادگیری پیوسته

تلاقی این دو حوزه، دقیقا محل کار حاضر است و چند مرجع کلیدی دارد. نخستین آن‌ها، CLPU [۱۶]، قوی‌ترین مرجع مبتنی بر جداسازی است. در پیاده‌سازی ارزیابی‌شده، هر وظیفه‌ی فعال شبکه‌ی کامل خود را دارد و یک درخواست حذف، همان مدخل را پاک می‌کند. در نتیجه تداخل بین‌وظیفه‌ای *به‌طور ساختاری* ناممکن است و بیشترین افت همیشه صفر است. هزینه‌ی این تضمین، رشد حالت مقیم با تعداد وظایف است. این پایان‌نامه CLPU را به‌عنوان سقف بالایی و نه رقیب در نظر می‌گیرد و در بخش ۴-۱۰ هزینه‌ی مقیم آن را به‌طور مستقیم حسابداری می‌کند.

PALL [۱۵] نقطه‌ی شروع فنی این کار است. PALL برای هر وظیفه یک زیرشبکه‌ی پراکنده می‌آموزد، ماسک آن را ذخیره می‌کند و در زمان درخواست، بر اساس اینکه یک مختصه در ماسک‌های تاریخی و

پشتیبان‌های ذخیره‌شده چه وضعیتی دارد، آن را بازنشانی یا بازسازی می‌کند و سپس وظایف نگه‌داری‌شده را ترمیم می‌کند. آنچه PALL فراهم می‌کند، ساختاری است که هم‌پوشانی را قابل‌اندازه‌گیری می‌سازد؛ آنچه فراهم نمی‌کند، تفکیک مختصه‌های مشترک بر حسب حساسیت نگه‌داری است.

نزدیک‌ترین کار به صورت‌مسئله‌ی ما یادگیری با فراموشی انتخابی[۳۳] [۵۵] است که یک کلاس را با کدهای یادیار پاک می‌کند در حالی که یادگیری ادامه دارد؛ همین جفت‌شدگی یادگیری و حذف در راهبرد پادگرا برای فراموشی انتخابی مادام‌العمر [۵۶] نیز تکرار می‌شود. تفاوت تنظیم ما با هر دو در آن است که یک مدل مشترک با حافظه‌ی بازپخش کران‌دار نگه می‌داریم و وزن‌های هم‌پوشان را مستقیما ویرایش می‌کنیم، نه آنکه فراموشی را از راه هدف یادگیری اعمال کنیم.

دو کار جدیدتر افق ارزیابی را گسترش می‌دهند. UnCLe [۵۷] پارامترهای مخصوص وظیفه را تولید می‌کند و به‌طور صریح بازگشت وظیفه پس از حذف[۳۴] را بدون داده‌ی نگه‌داری‌شده بررسی می‌کند. CATA [۵۸] حذف دنباله‌ای و ماندگاری آن را برای مدل‌های بینایی–زبان و از طریق تجمیع بردار وظیفه‌ی پرهیزگار از تعارض مطالعه می‌کند. [۵۹] نیز نشان می‌دهد تقطیر کنترل‌شده می‌تواند یادگیری و حذف را با حافظه‌ی ثابت هم‌زمان پشتیبانی کند. درس مشترک این کارها برای ما روش‌شناختی است: دقت بی‌درنگ پس از درخواست، شاهد کافی نیست و ماندگاری سرکوب باید جداگانه ممیزی شود. بخش ۴-۱۱-۵ همین ممیزی را انجام می‌دهد.

۲-۴ یادگیری کارآمد پارامتری

یادگیری کارآمد پارامتری[۳۵] آموزش را به ماژول‌های فشرده‌ی درج‌شده در یک شبکه‌ی منجمد محدود می‌کند. آداپترهای گلوگاهی[۳۶] [۶۰] یک تصویر پایین‌رو، یک غیرخطیت ReLU [۶۱] و یک تصویر بالارو را در قالب یک بلوک باقی‌مانده ترکیب می‌کنند:

$$h \leftarrow h + W_{up}\,\sigma\big(W_{down}\,h\big), \qquad W_{down} \in \mathbb{R}^{r\times d},\ W_{up} \in \mathbb{R}^{d\times r},\ r \ll d. \tag{۲-۳}$$

انطباق کم‌رتبه[۳۷] یا LoRA [۶۲] به‌جای درج ماژول، به‌روزرسانی وزن را به‌صورت حاصل‌ضرب دو ماتریس کم‌رتبه‌ی $\Delta W = BA$ پارامتری می‌کند. خط آداپترهای باقی‌مانده‌ی چندحوزه‌ای [۶۳]، زیرساخت‌های اشتراک آداپتر [۶۴]، نسخه‌های فشرده‌تر [۶۵، ۶۶] و روش‌های مبتنی بر پیشوند و پرامپت [۶۷، ۶۸]

[33] Learning with Selective Forgetting (LSF)
[34] Post-Deletion Task Relapse
[35] Parameter-Efficient Fine-Tuning (PEFT)
[36] Bottleneck Adapters
[37] Low-Rank Adaptation

این خانواده را تکمیل می‌کنند. پیش‌نویس اخیر BID-LoRA [۶۹] ترکیب مشابهی از یادگیری پیوسته و یادگیری‌زدایی را بررسی می‌کند.

انگیزه‌ی بررسی این خانواده در پایان‌نامه دو وجه دارد. وجه نخست کاهش دامنه‌ی به‌روزرسانی است: اگر بدنه منجمد باشد، درخواست حذف تنها بخش کوچکی از پارامترها را می‌بینند. وجه دوم مفهومی‌تر است: در یک آداپتر مشترک باریک، هم‌پوشانی میان وظایف به‌طور طبیعی متمرکز است، پس اگر سازوکار آگاه از هم‌پوشانی جایی به‌روشنی سود برساند، انتظار داریم آن‌جا باشد. فصل ۴ نشان می‌دهد این انتظار برآورده نمی‌شود و نتیجه‌ی منفی به‌صراحت گزارش می‌شود.

۲-۵ ارزیابی فراموشی و حریم خصوصی

۲-۵-۱ معیارهای رفتاری

سنجش کیفیت فراموشی معمولا با معیارهای رفتاری آغاز می‌شود: دقت روی وظیفه‌ی هدف پس از درخواست، دقت روی وظایف نگه‌داری‌شده، و اختلاف این دو نسبت به یک مرجع. نکته‌ی ظریفی که در ادبیات همیشه رعایت نمی‌شود این است که دقت وظیفه‌ی هدف را نباید با «کمتر بهتر» خواند: دقتِ صفر روی یک وظیفه‌ی دوکلاسه به این معناست که مدل با اطمینان *غلط* پاسخ می‌دهد و همین خودش یک سیگنال قابل بهره‌برداری است. خوانش درست، فاصله تا سطح تصادفی است. تمام جدول‌های فصل ۴ بر همین اساس تفسیر می‌شوند.

نکته‌ی ظریفی که در این معیارها اغلب نادیده گرفته می‌شود، جهت مطلوب دقت وظیفه‌ی حذف‌شده است. کاهش این دقت تا صفر مطلوب نیست، چون مدلی که روی وظیفه‌ی حذف‌شده *به‌طور نظام‌مند* اشتباه می‌کند، همچنان اطلاعاتی از آن وظیفه حمل می‌کند و یک حمله‌کننده می‌تواند از همین سوگیری برای تشخیص عضویت استفاده کند. هدف درست، رساندن دقت به سطح تصادفی است؛ یعنی مدل باید نسبت به وظیفه‌ی حذف‌شده *بی‌اطلاع* به نظر برسد و نه بدرفتار. به همین دلیل در این پایان‌نامه هرجا دقت وظیفه‌ی هدف گزارش می‌شود، معیار داوری فاصله‌ی آن تا سطح تصادفی است و نه مقدار خام آن.

۲-۵-۲ حمله‌ی استنتاج عضویت

حمله‌ی استنتاج عضویت[۳۸] [۱۰، ۷۰] تلاش می‌کند تشخیص دهد یک نمونه‌ی مشخص در آموزش مدل حاضر بوده است یا نه، و به همین دلیل به‌عنوان محک تجربی حریم خصوصی به‌کار می‌رود. نسخه‌ی

[38] Membership Inference Attack (MIA)

ساده‌ی آن از آستانه‌ی زیان استفاده می‌کند. دو هشدار مهم درباره‌ی این معیار وجود دارد. نخست، طراحی آزمون باید متقارن باشد: اگر نمونه‌های عضو با افزایش‌داده و نمونه‌های غیرعضو بدون آن پیش‌پردازش شوند، اختلاف مشاهده‌شده می‌تواند صرفا آرتیفکت پیش‌پردازش باشد. آزمایش‌های این پایان‌نامه به همین دلیل نسخه‌ی افزایش‌داده‌ی هم‌تراز را گزارش می‌کنند. دوم، حمله‌ی آستانه‌ی زیان کم‌توان است؛ حمله‌های قوی‌تر مبتنی بر مدل‌های سایه [۱۱] و ارزیابی‌های جامع‌تر [۷۱] می‌توانند تفکیکی نشان دهند که آزمون ساده از دست می‌دهد. بنابراین نتیجه‌ی نزدیک به ۰/۵ یک نتیجه‌ی تهیِ کم‌توان است و نه شاهد حذف.

خانواده‌ی حمله‌های دیگر نیز زمینه‌ی تهدید را کامل می‌کنند: وارون‌سازی مدل[۳۹] [۷۲] و استخراج داده‌ی آموزشی از مدل‌های زبانی [۷۳] نشان می‌دهند اثر داده در وزن‌ها تا چه اندازه قابل بازیابی است.

۲-۵-۳ مرجع بازآموزی و ممیزی صوری

یک ارزیابی دقیق‌تر، مقایسه‌ی مدل فراموش‌کرده با یک نمونه‌ی تازه از همان روش است که از ابتدا بدون وظیفه‌ی هدف آموزش دیده باشد. سنجه‌های مقایسه شامل توافق پیش‌بینی، فاصله‌ی جنسن–شانون[۴۰] میان توزیع‌های خروجی، فاصله‌ی ℓ_2 لاجیت‌ها و شباهت کسینوسی ویژگی‌ها است. در سوی صوری، [۲۰] روشی برای ممیزی حریم خصوصی با یک اجرای آموزش ارائه می‌دهد و حریم خصوصی تفاضلی[۴۱] [۵۳] چارچوب نظری استاندارد این حوزه است. بخش ۴-۱۱-۴ مرجع بازآموزی هم‌روش را برای EPALL گزارش می‌کند: نتیجه شاهدی برای هم‌ارزی با بازآموزی دقیق فراهم نمی‌کند، هرچند بدون یک خط پایه‌ی «بازآموزی در برابر بازآموزی» نمی‌توان از آن، حذف دقیق توزیعی را به‌طور صوری رد کرد.

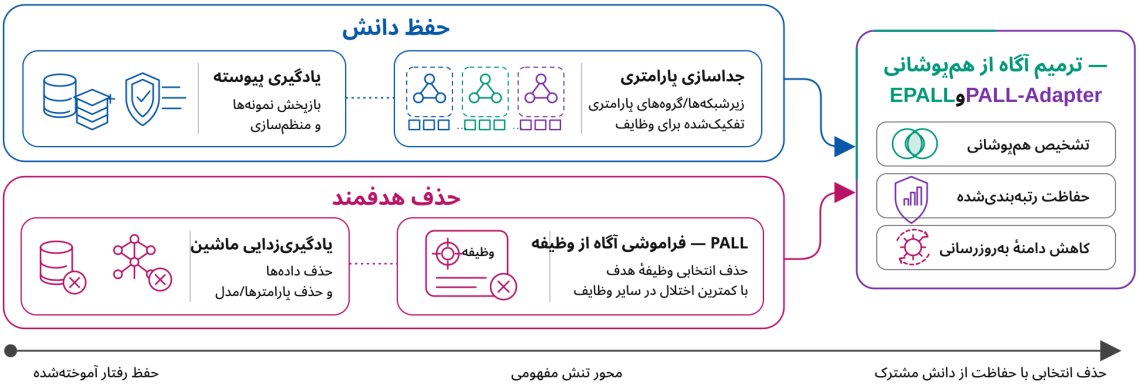
شکل ۲-۱ مسیر مفهومی ادبیات را از حفظ دانش گذشته تا حذف انتخابیِ آگاه از هم‌پوشانی خلاصه می‌کند. این شکل یک رده‌بندی مفهومی است و نه ترتیب تاریخی یا مقایسه‌ی تجربی.

۲-۶ مقایسه‌ی مفهومی و جایگاه این پژوهش

جدول ۲-۱ روش‌های مرتبط را بر چهار محور تصمیم‌گیری مقایسه می‌کند: آیا مدل میان وظایف مشترک است؟ آیا حذف در سطح وظیفه پشتیبانی می‌شود؟ آیا هم‌پوشانی پارامتری صریحا مدیریت می‌شود؟ و رشد حالت مقیم با تعداد وظایف چگونه است؟

خوانش این جدول، خلأ پژوهشی را روشن می‌کند. روش‌های تنظیم و بازپخش، سازوکار حذف در

[39] Model Inversion
[40] Jensen–Shannon
[41] Differential Privacy



رده‌بندی مفهومی؛ نه ترتیب تاریخی و نه نتیجهٔ تجربی

شکل ۲-۱: رده‌بندی مفهومی روش‌های مرتبط و جایگاه این پایان‌نامه. مسیر از روش‌های حفظ دانش در یادگیری پیوسته و جداسازی پارامتری، به یادگیری‌زدایی و سپس حذف وظیفه‌ایِ آگاه از هم‌پوشانی می‌رسد. این رده‌بندی برای سازمان‌دهی مفاهیم است و ادعای تجربی یا ترتیب تاریخی ندارد.

جدول ۲-۱: مقایسه‌ی مفهومی نزدیک‌ترین روش‌های مرتبط بر چهار محور تصمیم‌گیری این پایان‌نامه. دو سطر پایانی روش‌های پیشنهادی این کار هستند.

روش	مدل مشترک	حذف سطح وظیفه	مدیریت صریح هم‌پوشانی	رشد حالت مقیم
EWC [۷]	بله	خیر	خیر (جریمه‌ی سراسری)	ثابت
LwF [۳۲]	بله	خیر	خیر (تقطیر خروجی)	ثابت
ER / DER++ [۳۵]	بله	خیر	خیر	فقط حافظه
GEM / A-GEM [۳۷]	بله	خیر	ضمنی، تصویر سراسری	فقط حافظه
PackNet [۳۹]	بله	خیر	ماسک دارد، وزن منجمد	ماسک
WSN [۴۵]	بله	خیر	ماسک دارد، برای حذف به‌کار نمی‌رود	ماسک
SSD [۵۲]	بله	بله	فقط میراکردن اهمیتی	ثابت
SalUn [۵۱]	بله	بله	انتخاب وزن برای تغییر	ثابت
CLPU [۱۶]	خیر	بله	به‌طور ساختاری لازم نیست	یک شبکه به‌ازای وظیفه
PALL [۱۵]	بله	بله	ماسک ذخیره، مشترک‌ها بی‌رتبه	ماسک و پشتیبان
EPALL	بله	بله	**محافظت رتبه‌بندی‌شده‌ی مشترک**	ماسک و پشتیبان
PALL-Adapter	بله	بله	ماسک نرم (ممیزی‌شده، بی‌اثر)	آداپتر به‌ازای وظیفه

سطح وظیفه ندارند. روش‌های یادگیری‌زدایی عمومی مانند SSD و SalUn سازوکار حذف دارند اما ساختار وظیفه‌ای مدل را نمی‌بینند و — چنان‌که فصل ۴ نشان می‌دهد — در تنظیم پیوسته به سطح تصادفی وظیفه‌ی هدف نمی‌رسند. CLPU مسئله را با جداسازی کامل حل می‌کند و بنابراین مسئله‌ی هم‌پوشانی را از بین می‌برد نه اینکه حل کند؛ بهایش رشد خطی حالت مقیم است. روش‌های جداسازی پارامتری مانند WSN و PackNet ماسک‌ها را می‌سازند — یعنی همان چیزی که هم‌پوشانی ساختاری را خوش‌تعریف می‌کند — اما آن ماسک‌ها را برای عملیات حذف به‌کار نمی‌گیرند. PALL این دو را به هم می‌رساند اما مختصه‌های مشترک را رتبه‌بندی نمی‌کند.

خلأ باقی‌مانده، همان جای خالی است که این پایان‌نامه پر می‌کند: در یک سامانه‌ی *مدل‌مشترک* که ماسک‌های هر‌ـ‌وظیفه در اختیار دارد، هم‌پوشانی ساختاری را صریحا محاسبه کن، مختصه‌های مشترک را بر حسب حساسیت وظایف نگه‌داری‌شده رتبه‌بندی کن، و تنها زیرمجموعه‌ی بحرانی را در طول ترمیم مهار کن. فصل بعد این ایده را به یک الگوریتم مشخص تبدیل می‌کند.

فصل ۳

روش پیشنهادی

فصل پیشین نشان داد که مسئله‌ی فراموشی انتخابی در یک مدل مشترک، در نقطه‌ای گره می‌خورد که ماسک‌های هر ـ وظیفه در دست است اما از آن‌ها برای تفکیک مختصه‌های مشترک استفاده نمی‌شود. این فصل همان گره را باز می‌کند. ساختار آن چنین است: نخست مسئله رسما صورت‌بندی می‌شود و سازوکار ارثی PALL توضیح داده می‌شود؛ سپس تحلیل می‌شود که چرا یک به‌روزرسانی بی‌قید در ناحیه‌ی مشترک شکست می‌خورد؛ آنگاه روش اصلی این پایان‌نامه، EPALL، با تمام اجزا، الگوریتم و تحلیل پیچیدگی ارائه می‌شود؛ پس از آن مسیر اکتشافی PALL-Adapter به‌همراه یک تحلیل نظری با دامنه‌ی دقیقا مشخص‌شده معرفی می‌شود؛ و در پایان، پروتکل دسترسی به داده و آنچه این روش‌ها ادعا نمی‌کنند صریح می‌شود.

یادداشت نام‌گذاری. روش اصلی در این نوشتار EPALL نامیده می‌شود (سرنام Enhanced PALL). در کد پیاده‌سازی، لاگ‌های اجرا و نام برچسب‌های آزمایش، همین روش با شناسه‌ی pall_modified ثبت شده است؛ این دو نام به یک الگوریتم اشاره می‌کنند و تنها اختلاف نام‌گذاری میان متن و کد همین یک مورد است.

۳-۱ صورت‌بندی مسئله

۳-۱-۱ نمادگذاری

دنباله‌ی وظایف را با $\mathcal{T} = \{1, \ldots, n\}$ و مجموعه‌داده‌ی وظیفه‌ی t را با D_t نشان می‌دهیم؛ کلاس‌های وظایف مختلف مجزا هستند. شبکه با پارامترهای $\theta \in \mathbb{R}^d$ و یک سر طبقه‌بند به‌ازای هر وظیفه پیاده می‌شود. حافظه‌ی بازپخش $\mathcal{B}$ برای هر وظیفه‌ی t سه‌گانه‌ی (x, y, h) نگه می‌دارد که h لاجیت‌های ثبت‌شده در پایان

آموزش همان وظیفه است. سامانه دنباله‌ای از درخواست‌ها[1] را پردازش می‌کند؛ هر درخواست یا $\mathrm{learn}(t)$ است یا $\mathrm{forget}(t)$.

پس از دریافت $\mathrm{forget}(f)$، مجموعه‌ی وظایف فعال $\mathcal{T}_a = \{t \in \mathcal{T} : t$ آموزش‌دیده و حذف‌نشده $\} \setminus \{f\}$ است. دقت وظیفه‌ی t در لحظه‌ی r را با $A_t(r)$ و سطح تصادفی آن را با c_t نشان می‌دهیم. زیان نگه‌داری روی حافظه‌ی وظایف فعال چنین است:

$$\mathcal{L}_{\mathrm{retain}}(\theta) = \frac{1}{|\mathcal{T}_a|} \sum_{t \in \mathcal{T}_a} \mathbb{E}_{(x,y,h)\sim\mathcal{B}_t}\Big[\ell\big(f(x;\theta,t), y\big)\Big]. \tag{۳-۱}$$

۳-۱-۲ هدف مسئله

هدف عملیاتی درخواست فراموشی، یافتن پارامترهای جدید θ' است که هم‌زمان دو شرط را برآورده کند:

$$\big|A_f(\theta') - c_f\big| \to 0 \qquad \text{و} \qquad \max_{t \in \mathcal{T}_a}\big(A_t(\theta) - A_t(\theta')\big) \to 0. \tag{۳-۲}$$

شرط اول سرکوب هدف و شرط دوم حفظ بدترین وظیفه‌ی فعال است. عمدا از بیشینه و نه میانگین استفاده شده است: یک سامانه‌ی مستقر با میانگین خوب اما یک وظیفه‌ی ویران‌شده، از دید کاربران آن وظیفه شکست خورده است.

۳-۱-۳ سازوکار ارثی PALL

PALL [۱۵] برای هر وظیفه‌ی t یک ماسک دودویی $M_t \in \{0,1\}^d$ می‌آموزد که زیرشبکه‌ی همان وظیفه را مشخص می‌کند؛ این ماسک با امتیازهای یادگرفتنی[2] و در سطح پراکندگی هدف به‌دست می‌آید و پس از آموزش ذخیره می‌شود [۴۵، ۴۴]. استنتاج وظیفه‌ی t با $f(x;\theta \odot M_t, t)$ انجام می‌گیرد.

در زمان درخواست فراموشی، PALL سه کار انجام می‌دهد. یکم، ماسک M_f را از مجموعه‌ی فعال حذف می‌کند و مدخل بازپخش وظیفه‌ی هدف را پاک می‌کند. دوم، مختصه‌های ماسک هدف را بر اساس وضعیت آن‌ها در ماسک‌های تاریخی و پشتیبان‌های ذخیره‌شده *بازنشانی* یا *بازسازی* می‌کند. سوم، یک حلقه‌ی ترمیم روی حافظه‌ی وظایف فعال اجرا می‌کند تا افت جانبی جبران شود.

نکته‌ای که در گام دوم مهم است و در فصل ۴ به آن باز می‌گردیم این است که بازنشانی *شاخه‌محور* است. اینکه یک مختصه بازنشانی می‌شود یا حفظ می‌شود، به این بستگی دارد که وظیفه‌ی هدف پیش از

[1] Requests
[2] Learnable Scores

وظایف فعال آموزش دیده باشد یا پس از آن‌ها، و اینکه پشتیبان پراکنده‌ای برای آن مختصه ذخیره شده باشد یا نه. EPALL این منطق شاخه‌محور را دست‌نخورده به ارث می‌برد؛ نوآوری آن جای دیگری است.

۳-۲ محدودیت فراموشی بی‌قید

فرض کنید در پاسخ به درخواست، به‌جای سازوکار ساختاریافته، یک گام بی‌قید در جهت افزایش زیان وظیفه‌ی هدف برداشته شود:

$$\theta' = \theta + \eta\, \nabla_\theta \mathcal{L}_f(\theta). \tag{۳-۳}$$

اثر این گام بر یک وظیفه‌ی فعال t را می‌توان با بسط مرتبه‌ی اول تقریب زد:

$$\mathcal{L}_t(\theta') - \mathcal{L}_t(\theta) \;\approx\; \eta \left\langle \nabla_\theta \mathcal{L}_t(\theta),\; \nabla_\theta \mathcal{L}_f(\theta) \right\rangle. \tag{۴-۳}$$

عبارت سمت راست، ضرب داخلی گرادیان وظیفه‌ی فعال و گرادیان وظیفه‌ی هدف است. اگر این دو گرادیان همسو باشند، آسیب جانبی مثبت و بزرگ است. اما ضرب داخلی یک مقدار *انباشته* روی همه‌ی d مختصه است و همین انباشتگی، اطلاعات مهمی را پنهان می‌کند:

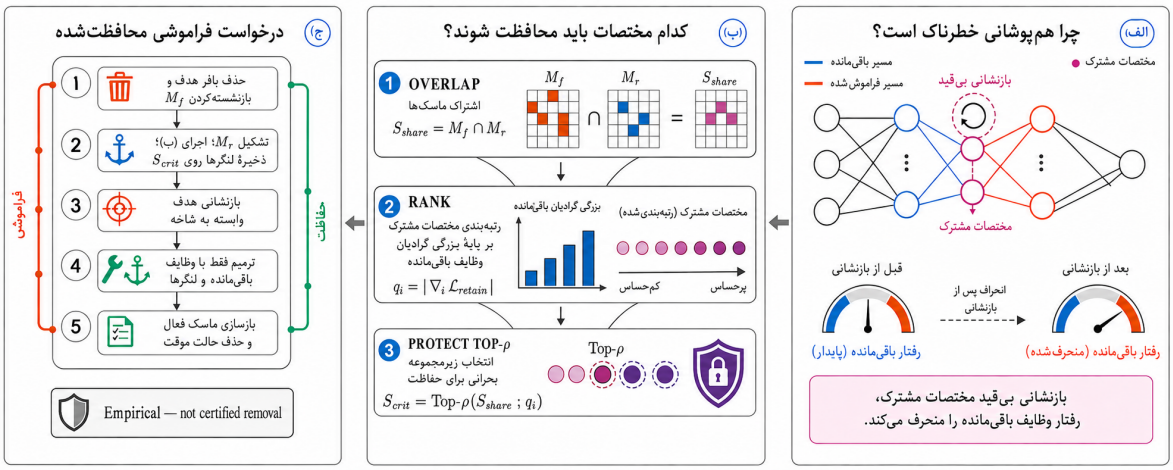
$$\left\langle \nabla \mathcal{L}_t,\; \nabla \mathcal{L}_f \right\rangle = \underbrace{\sum_{i \in S_{\text{share}}} \partial_i \mathcal{L}_t\, \partial_i \mathcal{L}_f}_{\text{ناحیه‌ی مشترک}} + \underbrace{\sum_{i \notin S_{\text{share}}} \partial_i \mathcal{L}_t\, \partial_i \mathcal{L}_f}_{\text{ناحیه‌ی غیرمشترک}}. \tag{۵-۳}$$

در جمله‌ی دوم، مختصه‌ها بیرون زیرشبکه‌ی وظایف فعال قرار دارند و بنابراین $|\partial_i \mathcal{L}_t|$ در آن‌ها کوچک است. پس *انتظار می‌رود* سهم آن جمله کوچک باشد و بخش عمده‌ی آسیب جانبی از جمله‌ی نخست بیاید. تأکید می‌شود که این یک استدلال ساختاری است و نه اندازه‌گیری: سهم نسبی دو جمله در این کار به‌طور تجربی تفکیک نشده است. مهم‌تر آنکه درون همان جمله‌ی نخست هم توزیع یکنواخت نیست: چند درصد از مختصه‌های مشترک، $|\partial_i \mathcal{L}_t|$ بزرگی دارند و بخش عمده‌ی مجموع را می‌سازند.

دو نتیجه‌ی طراحی از این تحلیل بیرون می‌آید. نتیجه‌ی اول، مهارکردن *همه‌ی* به‌روزرسانی —— آن‌گونه که یک تصویر سراسری در سبک GEM [۳۷] می‌کند —— لازم نیست و هزینه‌ی آن کندشدن فراموشی است. نتیجه‌ی دوم، مهارکردن همه‌ی ناحیه‌ی مشترک هم لازم نیست، چون بخش بزرگی از آن کم‌اهمیت است. راه‌حل درست، مهارکردن *حساس‌ترین* زیرمجموعه‌ی ناحیه‌ی مشترک است. این دقیقا کاری است که EPALL می‌کند و بخش ۴-۶ همین سه گزینه را به‌طور تجربی مقایسه می‌کند.

۳-۳ روش پیشنهادی EPALL

شکل ۳-۱ کل سازوکار را در سه پنل خلاصه می‌کند: مسئله، قاعده‌ی انتخاب، و رویه‌ی زمان درخواست. بخش‌های بعدی هر سه را باز می‌کنند.



شکل ۳-۱: سازوکار EPALL. الف) مسیر وظیفه‌ی فراموش‌شونده و مسیرهای وظایف نگه‌داری‌شده مختصه‌هایی را به اشتراک می‌گذارند، پس بازنشانی ساده رانش جانبی ایجاد می‌کند. ب) EPALL مجموعه‌ی $S_{\text{share}} = M_f \cap M_r$ را بر پایه‌ی حساسیت حافظه‌ی نگه‌داری‌شده رتبه‌بندی می‌کند و زیرمجموعه‌ی بالای ρ را محافظت می‌کند. ج) پس از تشخیص‌های اختیاری روی هدف، درخواست مدخل بازپخش هدف را حذف می‌کند، اجتماع ماسک‌های نگه‌داری‌شده را می‌سازد، لنگرها را ذخیره می‌کند، بازنشانی شاخه‌محور ارثی PALL را اعمال می‌کند و ترمیم لنگردار تنها روی وظایف نگه‌داری‌شده اجرا می‌شود. این‌ها اهداف تجربی‌اند، نه تضمین حذف گواهی‌شده.

۳-۳-۱ هم‌پوشانی ساختاری

M_f ماسک دودویی ذخیره‌شده‌ی وظیفه‌ی درخواست‌شده برای حذف و M_r اجتماع ماسک‌های وظایف نگه‌داری‌شده است:

$$M_r = \bigvee_{t \in \mathcal{T}_a} M_t. \tag{۳-۶}$$

پشتیبان مشترک[3] این دو، ناحیه‌ای است که هر تغییر در آن هم‌زمان به مسیر هدف و به حداقل یک مسیر فعال مربوط می‌شود:

$$S_{\text{share}} = M_f \cap M_r. \tag{۳-۷}$$

[3] Shared Support

اهمیت این تعریف در دقیق‌بودن آن است. چون ماسک‌ها دودویی و ذخیره‌شده‌اند، S_{share} یک اشتراک مجموعه‌ای است و به هیچ تقریب یا آستانه‌گذاری نیاز ندارد. این همان مزیتی است که ارث‌بری از خط زیرشبکه‌های ماسک‌دار [۴۵] فراهم می‌کند و در روش‌های بدون ماسک در دسترس نیست.

۳-۳-۲ رتبه‌بندی حساسیت و انتخاب بودجه

همه‌ی مختصه‌های S_{share} اهمیت یکسانی ندارند. برای هر $i \in S_{\text{share}}$ یک امتیاز حساسیت محاسبه می‌شود:

$$q_i = \left|\nabla_i \mathcal{L}_{\text{retain}}\right|, \tag{۳-۸}$$

که در آن گرادیان با یک گذر پس‌رو روی حافظه‌ی بازپخش وظایف فعال و روی زیرشبکه‌ی هر وظیفه محاسبه می‌شود. سپس با یک نسبت محافظت[۴] $\rho \in (۰, ۱]$، زیرمجموعه‌ی بحرانی انتخاب می‌شود:

$$S_{\text{crit}} = \Big\{ q_i \text{ مختصه‌ی برتر } S_{\text{share}} \text{ بر حسب } \lceil \rho\, |S_{\text{share}}| \rceil \Big\}. \tag{۳-۹}$$

مقدار پیش‌فرض $\rho = ۰/۲$ است. توجیه این انتخاب در بخش ۳-۲ آمد: سهم غالب آسیب جانبی از دنباله‌ی بالای توزیع q_i می‌آید، پس یک بودجه‌ی کوچک کافی است و بودجه‌ی بزرگ‌تر بی‌جهت فراموشی را کند می‌کند.

یک جانشین برای معادله‌ی (۳-۸) نیز پیاده‌سازی شده است: انرژی تعارض گرادیانی[۵]

$$q_i^{\text{conf}} = \text{ReLU}\big(- \nabla_i \mathcal{L}_f \cdot \nabla_i \mathcal{L}_{\text{retain}} \big), \tag{۳-۱۰}$$

که مختصه‌هایی را هدف می‌گیرد که فراموشی و نگه‌داری در آن‌ها مستقیما با هم رقابت می‌کنند. این جانشین به گرادیان وظیفه‌ی هدف نیاز دارد و بنابراین باید *پیش از* حذف مدخل بازپخش هدف محاسبه شود؛ همین قید، ترتیب گام‌های الگوریتم را تعیین می‌کند. بخش ۴-۱۲-۲ نشان می‌دهد نتیجه‌ی این جانشین وابسته به مجموعه‌داده است و بزرگی گرادیان پیش‌فرض می‌ماند.

۳-۳-۳ ترمیم لنگردار

پس از اعمال بازنشانی شاخه‌محور ارثی، ماسک ترمیم اسمی به‌صورت $M_f \cap M_r$ ساخته می‌شود و حلقه‌ی ترمیم تنها روی همان مختصه‌ها گام برمی‌دارد. تابع هدف ترمیم چنین است:

$$\mathcal{L}_{\text{repair}} = \alpha\, \mathcal{L}_{DER} + \beta\, \mathcal{L}_{CE} + \lambda \sum_{i \in S_{\text{crit}}} \big(\theta_i - \theta_i^{\text{anchor}}\big)^{۲}. \tag{۳-۱۱}$$

[4] Protection Ratio
[5] Gradient-Conflict Energy

سه جمله سه نقش متفاوت دارند. $\mathcal{L}_{DER}$ لاجیت‌های ذخیره‌شده را تطبیق می‌دهد و بنابراین *رفتار* پیشین مدل را بازمی‌گرداند [۳۵]. $\mathcal{L}_{CE}$ از برچسب‌های وظایف نگه‌داری‌شده استفاده می‌کند و سیگنال طبقه‌بندی مستقیم می‌دهد. جمله‌ی سوم، جمله‌ی لنگر[۶]، رانش را روی مختصه‌های بحرانی انتخاب‌شده محدود می‌کند. ضریب پیش‌فرض $\lambda = ۱$ است.

از نظر شکل ریاضی، جمله‌ی سوم همان قالب EWC [۷] است، اما دو تفاوت اساسی دارد که در بخش ۲-۲-۲ اشاره شد و اینجا صریح می‌شود: دامنه‌ی اعمال آن به S_{crit} محدود است و نه کل پارامترها، و هدف آن مهار یک عملیات *حذف* است و نه یک عملیات *یادگیری*.

انتخاب لنگر. دو گزینه برای θ^{anchor} وجود دارد. گزینه‌ی نخست θ^{old} است، یعنی مقدار وزن بلافاصله پیش از درخواست؛ این طبیعی‌ترین انتخاب برای هدف نگه‌داری است اما یک ایراد مفهومی جدی دارد: θ^{old} خودش در آموزش وظیفه‌ی فراموش‌شونده شکل گرفته و بنابراین اطلاعاتی از همان وظیفه حمل می‌کند. لنگرزدن یک عملیات *حذف* به مرجعی که خودش وابسته به داده‌ی حذف‌شونده است، از نظر مفهومی خودشکن است. گزینه‌ی دوم θ^{reinit} است، نمونه‌ای تازه از توزیع مقدارگذاری اولیه که به‌طور ساختاری از هر داده‌ای مستقل است.

این استدلال مفهومی به‌تنهایی θ^{reinit} را توصیه می‌کند، اما دو ملاحظه آن را تضعیف می‌کنند. نخست، لنگر تنها روی $S_{\text{crit}} \subseteq M_f \cap M_r$ عمل می‌کند، در حالی که فراموشی از راه بازنشانی ناحیه‌ی انحصاری $M_f \setminus M_r$ رخ می‌دهد؛ پس این جمله اهرم *نگه‌داری*است و نه اهرم فراموشی، و کشاندن مختصه‌های مشترک به سمت نوفه، باری را که همان مختصه‌ها برای وظایف نگه‌داری‌شده حمل می‌کنند تخریب می‌کند. دوم، θ^{reinit} استقلال از داده را کامل نمی‌کند، چون شاخه‌های بازنشانی ارثی همچنان از پشتیبان‌های پراکنده‌ی وابسته به داده‌ی قدیمی استفاده می‌کنند. مقایسه‌ی هشت‌بذری بخش ۴-۱۲-۱ همین را نشان می‌دهد: هیچ معیاری پس از اصلاح چندگانگی معنادار نمی‌شود و جهت شاهد روی CIFAR-100 به سود θ^{old} است. بنابراین θ^{old} پیش‌فرض پیاده‌سازی است و همه‌ی نتایج گزارش‌شده با همان تولید شده‌اند؛ ایراد مفهومی بالا در بخش ۴-۵ به‌عنوان محدودیت ثبت می‌شود.

۳-۳-۴ الگوریتم کامل

الگوریتم ۱ کل رویه‌ی زمان درخواست را گرد هم می‌آورد. ترتیب گام‌ها اتفاقی نیست: هر گرادیانی که به داده‌ی وظیفه‌ی هدف نیاز دارد، پیش از حذف مدخل بازپخش آن محاسبه می‌شود و پس از آن، هیچ بازگشتی به داده‌ی هدف وجود ندارد.

[6] Anchor

Algorithm 1 EPALL: request-time selective forgetting of task f

Input: weights θ, per-task masks $\{M_t\}$, rehearsal memory $\mathcal{B}$, active tasks $\mathcal{T}_a$, protection ratio ρ, anchor weight λ, repair steps K

Output: updated weights θ' with task f suppressed

copy M_f; if conflict scoring is selected, cache its forget gradient while $\mathcal{B}_f$ is available
delete $\mathcal{B}_f$; remove M_f from the active mask set ▷ no target-data access beyond this point
$M_r \leftarrow \bigvee_{t \in \mathcal{T}_a} M_t$; $S_{\text{share}} \leftarrow M_f \wedge M_r$
$q_i \leftarrow |\nabla_i \mathcal{L}_{\text{retain}}|$ for $i \in S_{\text{share}}$ ▷ one backward pass on $\mathcal{B}$
$S_{\text{crit}} \leftarrow$ top $\lceil \rho\, |S_{\text{share}}| \rceil$ coordinates of S_{share} by q_i
cache anchors $\theta^{\text{anchor}}|_{S_{\text{crit}}}$ and the sparse retained-task backups the inherited reset branches require
apply PALL's branch-dependent target reset/reconstruction
$M_{\text{repair}} \leftarrow M_f \wedge M_r$ ▷ nominal retained-repair mask
for $k = 1$ **to** K **do**
    sample retained rehearsal entries from $\mathcal{B}$
    $\mathcal{L} \leftarrow \alpha \mathcal{L}_{DER} + \beta \mathcal{L}_{CE} + \lambda \sum_{i \in S_{\text{crit}}} (\theta_i - \theta_i^{\text{anchor}})^2$
    backward; zero gradients outside M_{repair}; optimizer step
end for
rebuild the active union mask; discard request-only target state
return θ

سه ملاحظه‌ی پیاده‌سازی ارزش تأکید دارد. یکم، گام ۲ یک قید سخت است و نه یک توصیه؛ در پیاده‌سازی با یک ادعای اجرایی تضمین می‌شود که پس از حذف مدخل بازپخش هدف، هیچ مسیری به داده‌ی آموزشی خام یا مجموعه‌ی نگه‌داشته‌شده‌ی آن وظیفه بازنمی‌شود. دوم، گام ۶ تنها پشتیبان‌های پراکنده‌ای را ذخیره می‌کند که شاخه‌های بازنشانی ارثی به آن‌ها نیاز دارند، و در بخش ۴-۱۰ همین پشتیبان‌ها در حسابداری حالت مقیم شمرده می‌شوند. سوم، پس از پایان درخواست، حالت مخصوص درخواست —— از جمله لنگرها و ماسک آرشیوشده —— دور ریخته می‌شود، مگر آنکه اجرا در حالت ممیزی برون‌خط باشد.

۳-۳-۵ کنترل‌های مؤلفه‌ای

برای اینکه ادعای سازوکاری قابل آزمون باشد، چهار کنترل هم‌تراز پیاده‌سازی شده است که هر یک *تنها* یک جزء الگوریتم ۱ را عوض می‌کند و بقیه‌ی زمان‌بندی درخواست را دست‌نخورده می‌گذارد:

بدون جریمه‌ی ℓ_2. $\lambda = 0$. توجه شود که این کنترل *محافظت را برنمی‌دارد*: ناحیه‌ی بحرانی همچنان شناسایی می‌شود و پشتیبان‌های پراکنده‌ی آن نیز ساخته می‌شوند؛ تنها جمله‌ی جریمه‌ی ℓ_2 از تابع هدف حذف می‌شود. بنابراین نام درست آن «بدون جریمه‌ی ℓ_2» است و نه «بدون محافظت»، و آنچه جدا می‌کند سهم *شدت مهار* است و نه سهم کل سازوکار.

فقط هم‌پوشانی. کل S_{share} محافظت می‌شود، بدون هیچ رتبه‌بندی. این کنترل نشان می‌دهد آیا رتبه‌بندی حساسیت ارزشی افزون بر خودِ فیلتر ساختاری دارد.

بودجه‌ی تصادفی. همان تعداد مختصه‌ی کنترل کامل محافظت می‌شود، اما به‌طور تصادفی یکنواخت از میان S_{share} انتخاب می‌شود. این کنترل، اثر *اندازه‌ی* بودجه را از اثر *کیفیت* انتخاب جدا می‌کند. توجه شود که نمونه‌گیری از S_{share} انجام می‌شود و نه از M_f؛ اگر از M_f نمونه گرفته می‌شد، مختصه‌های منحصر به هدف در بودجه وارد می‌شدند که گرادیان‌شان بعدا در ترمیم ماسک می‌شود و در نتیجه بودجه‌ی مؤثر کنترل از کنترل کامل کوچک‌تر می‌شد.

رتبه‌بندی بدون هم‌پوشانی. رتبه‌بندی با همان q_i روی کل زیرشبکه‌ی هدف انجام می‌شود، بدون اعمال صریح فیلتر M_r. این کنترل می‌سنجد که آیا اشتراک صریح اطلاعاتی افزون بر رتبه‌بندی دارد.

نتایج این چهار کنترل در بخش ۴-۶ گزارش می‌شود و — چنان‌که خواهیم دید — یکی از آن‌ها نتیجه‌ای می‌دهد که ساده‌سازی روش را پیشنهاد می‌کند.

۳-۳-۶ تحلیل پیچیدگی

هزینه‌ی افزوده‌ی EPALL نسبت به PALL اصلی سه جزء دارد. جزء اول، محاسبه‌ی q_i است که یک گذر پس‌رو روی حافظه‌ی بازپخش وظایف فعال لازم دارد؛ هزینه‌ی آن $\mathcal{O}(|\mathcal{B}| \cdot C_{\text{bwd}})$ است که C_{bwd} هزینه‌ی یک گذر پس‌رو است. چون $|\mathcal{B}| = 500$ و ثابت است، این جزء با تعداد وظایف رشد نمی‌کند. جزء دوم، انتخاب $\lceil \rho |S_{\text{share}}| \rceil$ مختصه‌ی برتر است که با انتخاب مرتبه‌ای در $\mathcal{O}(|S_{\text{share}}|)$ یا با مرتب‌سازی در $\mathcal{O}(|S_{\text{share}}| \log |S_{\text{share}}|)$ انجام می‌شود و در هر حال از $\mathcal{O}(d)$ فراتر نمی‌رود. جزء سوم، ارزیابی جمله‌ی لنگر در هر گام ترمیم است که $\mathcal{O}(|S_{\text{crit}}|)$ و بنابراین حداکثر $\mathcal{O}(\rho\, d)$ است.

از نظر حافظه، EPALL علاوه بر ماسک‌های PALL، مقادیر لنگر روی S_{crit} را نگه می‌دارد که $\mathcal{O}(\rho|S_{\text{share}}|)$ است و پس از پایان درخواست آزاد می‌شود.

مجموع این هزینه‌ها در تئوری کوچک است، اما در عمل به‌صورت افزایش زمان درخواست دیده می‌شود، چون گذر پس‌روی اضافی و جمله‌ی لنگر در هر یک از K گام ترمیم ارزیابی می‌شوند. بخش ۳-۴ این هزینه را به‌طور صریح گزارش می‌کند: EPALL روی CIFAR-10 حدود ۲/۱ ثانیه و روی CIFAR-100 حدود ۱۲/۱ ثانیه از PALL اصلی کندتر است. این یک مصالحه‌ی واقعی است و پنهان نمی‌شود.

۳-۴ مسیر اکتشافی PALL-Adapter

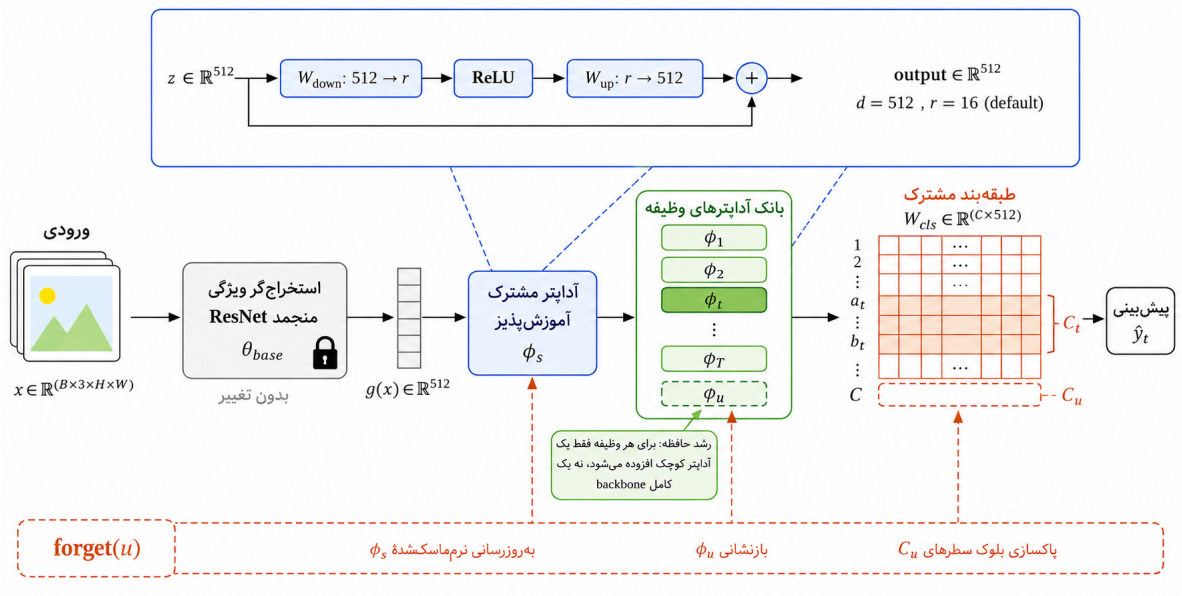
بخش‌های پیشین روش اصلی را کامل کردند. این بخش یک مسیر دوم را معرفی می‌کند که با انگیزه‌ی کاهش دامنه‌ی به‌روزرسانی طراحی شد و در نهایت به یک *نتیجه‌ی منفی* رسید. ارائه‌ی آن دو دلیل دارد: نخست، طراحی و ممیزی آن بخشی از کار انجام‌شده است؛ دوم، شکل ممیزی آن — پنج بازوی هم‌تراز که هر یک یک مؤلفه را حذف می‌کند — الگویی است که می‌تواند در کارهای مشابه به‌کار رود.

۳-۴-۱ معماری

لایه‌های استخراج ویژگی θ_{base} منجمد می‌شوند و تنها سه گروه پارامتر آموزش‌پذیر می‌مانند: یک آداپتر گلوگاهی مشترک[۷] با پارامترهای ϕ_s، یک آداپتر مخصوص وظیفه[۸] با پارامترهای ϕ_t به‌ازای هر وظیفه، و یک طبقه‌بند مشترک که از طریق برش سطری مخصوص وظیفه استفاده می‌شود. هر دو آداپتر، بلوک‌های باقی‌مانده‌ی گلوگاهی استاندارد پس از بدنه با غیرخطیت ReLU هستند [۶۰، ۶۱] و به‌ترتیب اعمال می‌شوند: نخست مشترک، سپس مخصوص وظیفه. شکل ۳-۲ این ساختار را نشان می‌دهد.

مقدارگذاری اولیه‌ی آداپترها به‌گونه‌ای است که در لحظه‌ی تولد، ماژول تابع همانی باشد: تصویر پایین‌رو از توزیع کایمینگ [۳۱] نمونه می‌گیرد و تصویر بالارو با صفر مقدار می‌گیرد. در نتیجه درج آداپتر به یک شبکه‌ی آموزش‌دیده، رفتار آن را در ابتدا تغییر نمی‌دهد.

[7] Shared Bottleneck Adapter
[8] Task-Specific Adapter



شکل ۳-۲: معماری PALL-Adapter. بدنه‌ی استخراج ویژگی منجمد است. جریان اطلاعات از آداپتر مشترک ϕ_s می‌گذرد، سپس به آداپتر مخصوص وظیفه‌ی ϕ_t می‌رسد و در پایان از برش سطری متناظر در طبقه‌بند مشترک عبور می‌کند. تنها همین سه گروه در آموزش و در درخواست فراموشی تغییر می‌کنند.

۳-۴-۲ فراموشی سه‌فازی

درخواست فراموشی در این مسیر ترتیب سخت‌گیرانه‌ای دارد تا هر گرادیان وابسته به هدف، در زمانی برآورد شود که مدخل بازپخش هدف هنوز موجود است.

فاز یک: کمی‌سازی هم‌پوشانی. گرادیان‌های قدرمطلق آداپتر مشترک روی حافظه‌ی هدف و حافظه‌ی نگه‌داری‌شده محاسبه می‌شوند و با نسبت‌های آستانه‌ای، دو مجموعه تعریف می‌شود: S_{forget} شامل مختصه‌های با بالاترین گرادیان هدف، و S_{active} شامل مختصه‌های با بالاترین گرادیان نگه‌داری. اشتراک آن‌ها ناحیه‌ی بحرانی است:

$$S_{\text{crit}} = S_{\text{forget}} \cap S_{\text{active}}, \tag{۳-۱۲}$$

و $\lceil \alpha_p |S_{\text{crit}}| \rceil$ مختصه‌ی برتر آن بر حسب بزرگی گرادیان نگه‌داری، مجموعه‌ی سفت‌محافظت‌شده[9] H را می‌سازد. توجه شود که این مجموعه‌ها با مجموعه‌های ماسک‌محور بخش ۳-۳ متفاوت‌اند: آن‌ها از ماسک‌های دودویی ذخیره‌شده می‌آیند و این‌ها از رتبه‌بندی گرادیان.

فاز دو: بازنشانی هدف و به‌روزرسانی ماسک‌نرم. آداپتر مخصوص وظیفه‌ی هدف با همان قاعده‌ی مقدارگذاری اولیه بازنشانی می‌شود و برش طبقه‌بند آن پاک می‌شود. سپس آداپتر مشترک در جهت رسیدن

[9] Hard-Protected

به توزیع یکنواخت روی کلاس‌های وظیفه‌ی هدف بهینه می‌شود، با این تفاوت که گام هر مختصه در یک ضریب مقیاس ضرب می‌شود:

$$m_{\text{soft}}(i) = \begin{cases} ۰, & i \in H, \\ ۱ - p, & i \in S_{\text{crit}} \setminus H, \\ ۱, & i \in S_{\text{forget}} \setminus S_{\text{active}}, \\ ۰, & \text{در غیر این صورت،} \end{cases} \tag{۱۳-۳}$$

که $p \in [۰, ۱]$ شدت محافظت است. منطق این چهار حالت روشن است: ناحیه‌ی سفت‌محافظت‌شده تکان نمی‌خورد؛ باقی ناحیه‌ی بحرانی با ضریب کاهش‌یافته حرکت می‌کند؛ ناحیه‌ی منحصر به هدف آزادانه به‌روز می‌شود؛ و مختصه‌های بیرون S_{forget} اصلا هدف این فاز نیستند.

فاز سه: صعود طبقه‌بند و ترمیم. برش طبقه‌بند وظیفه‌ی هدف یک گام صعود گرادیانی از گرادیان فراموشی *پیش از بازنشانی* خود می‌گیرد، مدخل بازپخش هدف حذف می‌شود، و همه‌ی پارامترهای آموزش‌پذیر برای چند گام کوتاه روی حافظه‌ی وظایف نگه‌داری‌شده ترمیم می‌شوند.

۳-۴-۳ دامنه‌ی واقعی ماسک نرم

نکته‌ی مهمی که پیش از هر تحلیل نظری باید تصریح شود، این است: ضریب m_{soft} تنها جابه‌جایی آداپتر مشترک را در *طول زیرفاز هدف*ــ *یکنواخت* کنترل می‌کند. این ضریب یک ماسک محافظت دائمی نیست و دقت نگه‌داری انتهابه‌انتها را کران نمی‌زند، چون سه عامل دیگر هم نقطه‌ی پایان را جابه‌جا می‌کنند: بازنشانی آداپتر هدف، صعود گرادیانی طبقه‌بند، و ترمیم پایانی که ماسک‌نشده است. علاوه بر این، چون آداپتر مشترک نهایی تابعی از گرادیان‌های حافظه‌ی هدف می‌ماند، این مسیر نه فراموشی دقیق است و نه گواهی‌شده از نظر حریم خصوصی.

۳-۵ تحلیل نظری با دامنه‌ی محدود

این بخش یک گزاره‌ی دقیق درباره‌ی همان چیزی اثبات می‌کند که ماسک نرم واقعا کنترل می‌کند: تغییر مخاطره‌ی وظایف فعال در *طول زیرفاز ماسک‌نرم*. سپس بخش ۳-۵-۴ توضیح می‌دهد چرا این گزاره به افت انتهابه‌انتها تعمیم نمی‌یابد، و چرا این محدودیت با یافته‌ی تجربی فصل ۴ سازگار است.

۳-۵-۱ نمادگذاری و افراز

در سراسر این تحلیل، بدنه‌ی منجمد θ_{base} و آداپترهای مخصوص وظیفه ثابت‌اند؛ پس تنها پارامترهای متغیر و در نتیجه تنها منبع تداخل میان‌وظیفه‌ای در این زیرفاز، وزن‌های آداپتر مشترک $\phi \triangleq \phi_s \in \mathbb{R}^{d_s}$ هستند. $\phi^{(۰)}$ مقدار آن در آغاز زیرفاز و $\phi^{(N)}$ مقدار آن پس از N گام است. به‌روزرسانی هر مختصه چنین است:

$$\phi_i^{(k+۱)} = \phi_i^{(k)} - \eta\, m_i\, g_i^{(k)}, \qquad g_i^{(k)} \triangleq \partial_i\, \mathcal{L}_{\text{unlearn}}\big(\phi^{(k)}\big), \tag{۳-۱۴}$$

که $m_i = m_{\text{soft}}(i) \in [۰, ۱]$ در طول N گام ثابت است، چون ماسک یک‌بار در فاز یک محاسبه می‌شود. مجموعه‌ی اندیس‌ها به چهار ناحیه‌ی مجزا افراز می‌شود:

$$\begin{aligned}
\mathcal{F}^{\circ} &\triangleq S_{\text{forget}} \setminus S_{\text{active}} && (m_i = ۱),\\
\mathcal{S} &\triangleq S_{\text{crit}} \setminus H && (m_i = ۱ - p),\\
H &\subseteq S_{\text{crit}} && (m_i = ۰),\\
\mathcal{N} &\triangleq \{۱, \dots, d_s\} \setminus S_{\text{forget}} && (m_i = ۰).
\end{aligned} \tag{۳-۱۵}$$

تعریف ۳-۱ (مخاطره‌ی نگه‌داری و افت زیرفاز) *برای هر وظیفه‌ی فعال $t \in \mathcal{T}_a$، مخاطره‌ی آن را به‌صورت تابعی از آداپتر مشترک با $R_t(\phi) \triangleq \mathbb{E}_{(x,y)\sim D_t}[\ell(f(x;\phi), y)]$ تعریف می‌کنیم. افتی که مسیر زیرفاز $\phi^{(۰)} \to \phi^{(N)}$ بر وظیفه‌ی t تحمیل می‌کند برابر $\Delta_t \triangleq R_t(\phi^{(N)}) - R_t(\phi^{(۰)})$ است. تأکید می‌شود که Δ_t افت همین زیرفاز است و نه بیشترین افت گزارش‌شده در فصل ۴.*

۳-۵-۲ فرض‌ها

فرض ۳-۱ (همواری مخاطره‌ی نگه‌داری) *هر R_t روی پوشش محدب مسیر، βـ هموار است: $\|\nabla R_t(\phi) - \nabla R_t(\phi')\|_۲ \leqslant \beta\|\phi - \phi'\|_۲$.*

فرض ۳-۲ (کران‌داری گرادیان‌های فراموشی) *$|g_i^{(k)}| \leqslant G$ به‌طور یکنواخت روی مسیر؛ در نتیجه جابه‌جایی انباشته‌ی هر مختصه از $\Gamma_i \triangleq |\sum_{k=۰}^{N-۱} g_i^{(k)}| \leqslant NG$ پیروی می‌کند.*

فرض ۳-۳ (تمرکز حساسیت نگه‌داری روی ناحیه‌ی مشترک) *$\epsilon \geqslant ۰$ای وجود دارد که $|\partial_i R_t(\phi^{(۰)})| \leqslant \epsilon$ برای هر $i \in \mathcal{F}^{\circ}$ و هر $t \in \mathcal{T}_a$.*

فرض ۳-۳ یک *فرض اضافی* درباره‌ی تک‌تک مخاطره‌های نگه‌داری است و از قاعده‌ی انتخاب الگوریتم به‌تنهایی نتیجه نمی‌شود. الگوریتم مختصه‌ها را با بزرگی گرادیان تجمیعی وظایف نگه‌داری رتبه‌بندی می‌کند؛ در نتیجه، خنثی‌شدن گرادیان‌های وظایف مختلف می‌تواند یک مختصه با مشتق بزرگ برای یک وظیفه‌ی منفرد را بیرون مجموعه‌ی انتخابی نگه دارد. بنابراین لم بعدی مشروط به برقرار بودن این تمرکز است و اگر کوچک‌ترین ϵ معتبر بزرگ باشد، کران می‌تواند تهی شود.

۳-۵-۳ کران بالای افت زیرفاز و یکنوایی آن نسبت به p

لم ۳-۴ (کران بالای افت زیرفاز) *فرض‌های ۳-۱ تا ۳-۳ را برقرار بگیرید و $\eta > 0$ و N را ثابت کنید. برای هر وظیفه‌ی فعال t، انرژی تداخل بحرانی[10] را چنین تعریف کنید:*

$$E_t^{\mathrm{crit}} \triangleq \eta \sum_{i \in S_{\mathrm{crit}}} \left|\partial_i R_t(\phi^{(0)})\right| \Gamma_i, \tag{۳-۱۶}$$

و انرژی پس‌مانده‌ی منحصر به هدف را $C_t \triangleq \eta \sum_{i \in \mathcal{F}^\circ} \Gamma_i$. آنگاه به‌روزرسانی ماسک‌نرم با شدت محافظت p در کران زیر صدق می‌کند:

$$\Delta_t^{\mathrm{soft}} \leqslant (1-p)\, E_t^{\mathrm{crit}} + \epsilon\, C_t + \tfrac{\beta}{2}\, \eta^2 N^2 G^2 \Big(|\mathcal{F}^\circ| + (1-p)^2 |\mathcal{S}|\Big), \tag{۳-۱۷}$$

حال‌آنکه به‌روزرسانی بی‌قید ($m_i = 1$ روی کل S_{forget}) از کران زیر پیروی می‌کند:

$$\Delta_t^{\mathrm{unc}} \leqslant E_t^{\mathrm{crit}} + \epsilon\, C_t + \tfrac{\beta}{2}\, \eta^2 N^2 G^2\, |S_{\mathrm{forget}}|. \tag{۳-۱۸}$$

سمت راست (۳-۱۷) را کران جانشین می‌نامیم و با $B_t(p)$ نشان می‌دهیم.

اثبات. وظیفه‌ی فعال t را ثابت می‌گیریم و جابه‌جایی کل آداپتر مشترک را $\delta \triangleq \phi^{(N)} - \phi^{(0)}$ می‌نامیم. با جمع‌بستن (۳-۱۴) روی N گام و ثابت‌بودن m_i:

$$\delta_i = -\eta\, m_i \sum_{k=0}^{N-1} g_i^{(k)} \qquad \Longrightarrow \qquad |\delta_i| \leqslant \eta\, m_i\, \Gamma_i \leqslant \eta\, m_i\, NG, \tag{۳-۱۹}$$

که از نابرابری مثلثی و فرض ۳-۲ می‌آید.

گام ۱ (بسط همواری). طبق فرض ۳-۱، R_t کران بالای درجه‌دوم زیر را در نقطه‌ی پایه‌ی $\phi^{(0)}$ می‌پذیرد:

$$\Delta_t \leqslant \big\langle \nabla R_t(\phi^{(0)}),\, \delta \big\rangle + \tfrac{\beta}{2}\, \|\delta\|_2^2. \tag{۳-۲۰}$$

[10] Critical Interference Energy

گام ۲ (جمله‌ی مرتبه‌ی اول). با افراز (۳-۱۵) و نابرابری مثلثی:

$$\begin{aligned}\langle \nabla R_t, \delta \rangle &\leqslant \sum_{i=1}^{d_s} \left|\partial_i R_t\right| |\delta_i| \\ &\leqslant \underbrace{\sum_{i\in\mathcal{F}^\circ} \left|\partial_i R_t\right| \eta\, \Gamma_i}_{m_i=1} + \underbrace{\sum_{i\in\mathcal{S}} \left|\partial_i R_t\right| (1-p)\, \eta\, \Gamma_i}_{m_i=1-p} + \underbrace{\sum_{i\in H\cup\mathcal{N}} \left|\partial_i R_t\right| \cdot 0}_{m_i=0}. \end{aligned} \tag{۳-۲۱}$$

جمله‌ی سوم هویتا صفر است. جمله‌ی نخست طبق فرض ۳-۳ حداکثر $\epsilon\eta\sum_{i\in\mathcal{F}^\circ}\Gamma_i = \epsilon C_t$ است. جمله‌ی دوم، چون $\mathcal{S} \subseteq S_{\text{crit}}$ و همه‌ی جملاتش نامنفی‌اند، حداکثر $(1-p)E_t^{\text{crit}}$ است. پس

$$\langle \nabla R_t, \delta \rangle \leqslant (1-p)\, E_t^{\text{crit}} + \epsilon\, C_t. \tag{۳-۲۲}$$

گام ۳ (جمله‌ی مرتبه‌ی دوم). از (۳-۱۹)، $|\delta_i|^2 \leqslant \eta^2 m_i^2 N^2 G^2$؛ و چون $m_i^2 = 1$ روی $\mathcal{F}^\circ$، $m_i^2 = (1-p)^2$ روی $\mathcal{S}$، و $m_i = 0$ در دو ناحیه‌ی دیگر:

$$\tfrac{\beta}{2}\|\delta\|_2^2 \leqslant \tfrac{\beta}{2}\,\eta^2 N^2 G^2 \Big(|\mathcal{F}^\circ| + (1-p)^2 |\mathcal{S}|\Big). \tag{۳-۲۳}$$

جای‌گذاری (۳-۲۲) و (۳-۲۳) در (۳-۲۰)، کران (۳-۱۷) را می‌دهد.

گام ۴ (خط پایه‌ی بی‌قید). برای به‌روزرسانی بی‌قید، $m_i = 1$ روی $S_{\text{forget}} = \mathcal{F}^\circ \cup S_{\text{crit}}$ و $m_i = 0$ روی $\mathcal{N}$. تکرار گام‌های ۲ و ۳ با این ضرب‌گرها، جمله‌ی بحرانی را بدون مقیاس‌دهی و جمله‌ی انحنا را برابر $\frac{\beta}{2}\eta^2 N^2 G^2 |S_{\text{forget}}|$ می‌دهد؛ یعنی دقیقا (۳-۱۸).

□

نتیجه‌ی ۳-۵ (یکنوایی کران جانشین نسبت به شدت محافظت) *کران جانشین* $B_t(p) = (1-p)E_t^{\text{crit}} + \epsilon C_t + \frac{\beta}{2}\eta^2 N^2 G^2\big(|\mathcal{F}^\circ| + (1-p)^2|\mathcal{S}|\big)$ *روی* $p \in [0,1]$ *مشتق‌پذیر و ناافزاینده است:*

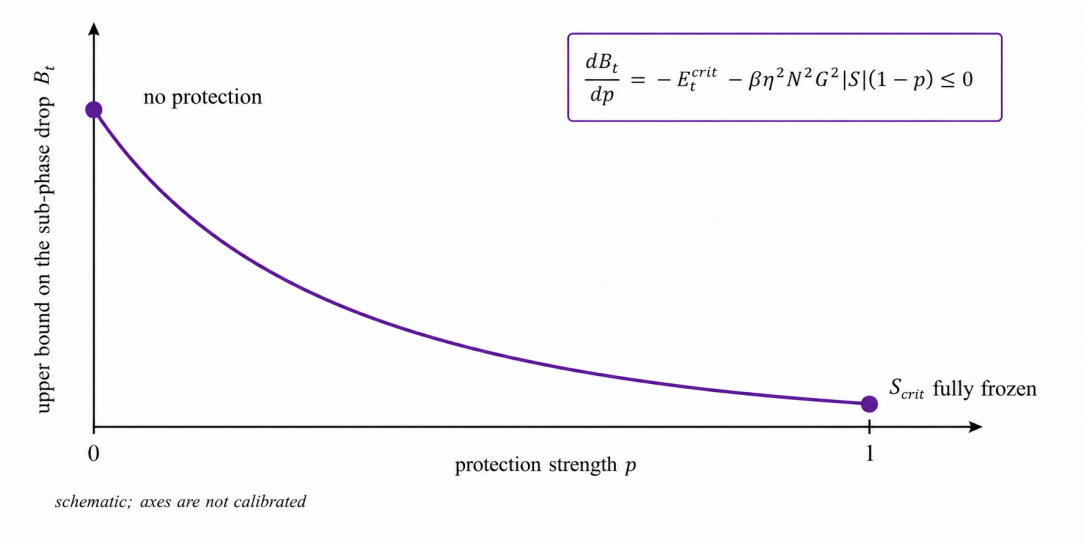
$$\frac{\mathrm{d}B_t}{\mathrm{d}p} = -\,E_t^{\text{crit}} \;-\; \beta\,\eta^2 N^2 G^2\,|\mathcal{S}|\,(1-p) \;\leqslant\; 0. \tag{۳-۲۴}$$

پس با بزرگ‌شدن p کران کوچک‌تر می‌شود و نرخ کاهش آن دست‌کم E_t^{crit} است. در $p = 1$ — یا هم‌ارز، با گسترش H تا پوشش کل S_{crit} — سهم ناحیه‌ی بحرانی از کران کاملا حذف می‌شود و تنها ϵC_t و جمله‌ی انحنای محدود به $\mathcal{F}^\circ$ باقی می‌ماند. افزون بر این $B_t(0) \leqslant B_t^{\text{unc}}$، با تساوی دقیقا وقتی $H = \varnothing$.

مشاهده‌ی ۳-۶ (آنچه از این کران نتیجه نمی‌شود) *از $\Delta_t^{\text{soft}} \leqslant B_t(p)$ و $\Delta_t^{\text{unc}} \leqslant B_t^{\text{unc}}$ و $B_t(p) \leqslant B_t^{\text{unc}}$ نمی‌توان نتیجه گرفت که $\Delta_t^{\text{soft}} \leqslant \Delta_t^{\text{unc}}$ یا $\Delta_t^{\text{soft}} \leqslant (1-p)\Delta_t^{\text{unc}}$. مقایسه‌ی دو کران بالا هیچ*

رابطه‌ای میان دو مقدار واقعی برقرار نمی‌کند: هر دو مقدار می‌توانند به‌اندازه‌ی دلخواه از کران خود پایین‌تر باشند و ترتیبشان وارونه‌ی ترتیب کران‌ها شود. بنابراین ادعای این بخش تنها درباره‌ی پوش بالای نظری است و نه درباره‌ی افت واقعی.

شکل ۳-۳ یکنوایی این کران جانشین نسبت به p را نشان می‌دهد. حال یک پرسش طبیعی مطرح می‌شود: آیا می‌توان کران (۳-۱۷) را با لاگ‌های اجرا آزمود؟ پاسخ منفی است: مقدار محاسبه‌شده‌ی طرف راست کران و افت اندازه‌گیری‌شده یکای یکسان ندارند؛ اولی در یکای مخاطره و دومی در یکای دقت است، و ثابت‌های ϵ و β نیز کالیبره نشده‌اند. بنابراین یک نمودار پراکندگی میان این دو، حتی اگر هم‌بستگی مرتبه‌ای نشان دهد، نمی‌تواند صحت‌سنجی کران باشد و در این پایان‌نامه هیچ ادعایی بر پایه‌ی چنین تشخیصی بنا نمی‌شود.



شکل ۳-۳: کران بالای B_t افت یک وظیفه‌ی فعال در زیرفاز ماسک‌نرم، بر حسب شدت محافظت p. مطابق نتیجه‌ی ۳-۵، مشتق dB_t/dp در سراسر بازه نامثبت است، پس کران هرگز افزایش نمی‌یابد؛ چون قدر مطلق شیب با بزرگ‌شدن p کاهش می‌یابد، منحنی نزولی و محدب است. محورها کالیبره نشده‌اند و شکل تنها جهت را نشان می‌دهد. تأکید می‌شود که این گزاره درباره‌ی *کران* است و نه درباره‌ی افت واقعی؛ مشاهده‌ی ۳-۶ همین تمایز را تصریح می‌کند.

۳-۵-۴ چرا این کران، افت انتهابه‌انتها را کران نمی‌زند

لم ۳-۴ درست است، اما دامنه‌ی آن باریک است و این باریکی باید صریح بماند. سه شکاف میان Δ_t در تعریف ۳-۱ و بیشترین افت گزارش‌شده در فصل ۴ وجود دارد:

۱. **بازنشانی پیش از زیرفاز.** آداپتر مخصوص وظیفه‌ی هدف و برش طبقه‌بند آن *پیش از* شروع زیرفاز

ماسک‌نرم بازنشانی می‌شوند. این جابه‌جایی در $\phi^{(\cdot)}$ گنجانده شده است، پس تحلیل درباره‌ی اثر آن هیچ نمی‌گوید — در حالی که همان اثر می‌تواند بخش عمده‌ی تغییر نقطه‌ی پایان باشد.

۲. **صعود طبقه‌بند.** گام صعود گرادیانی روی برش طبقه‌بند هدف، بیرون از فضای ϕ_s عمل می‌کند و ماسک نرم آن را نمی‌بیند.

۳. **ترمیم ماسک‌نشده.** فاز سه همه‌ی پارامترهای آموزش‌پذیر — از جمله مختصه‌های H — را روی حافظه‌ی وظایف نگه‌داری‌شده به‌روز می‌کند. بنابراین حتی مختصه‌هایی که در زیرفاز قبل قفل بودند، در نقطه‌ی پایان جابه‌جا شده‌اند.

نتیجه‌ی این سه شکاف آن است که یکنوایی کران جانشین نسبت به p هیچ تضمینی روی معیار گزارش‌شده نمی‌سازد. به همین دلیل، این پایان‌نامه لم ۳-۴ را به‌عنوان توضیح سازوکار زیرفاز ارائه می‌کند و نه به‌عنوان پشتوانه‌ی نظری برای نتایج تجربی، و اثر واقعی ماسک نرم را با کنترل‌های مؤلفه‌ای می‌سنجد. بخش ۴-۸ نشان می‌دهد آن کنترل‌ها هیچ سود قابل‌اندازه‌گیری برای ماسک نرم در نقطه‌ی پایان پیدا نمی‌کنند و سرکوب مشاهده‌شده را به بازنشانی مسیر هدف نسبت می‌دهند. این هم‌نشینی — یک کران درست با دامنه‌ی باریک، در کنار یک نتیجه‌ی تجربی منفی برای دامنه‌ی وسیع — تعارض نیست؛ دقیقا همان چیزی است که از تفکیک دقیق دامنه‌ها انتظار می‌رود.

۳-۶ پروتکل دسترسی به داده

هر دو مسیر پیشنهادی یک قید مشترک را رعایت می‌کنند که برای اعتبار ادعاها حیاتی است: هر گرادیانی که به داده‌ی وظیفه‌ی هدف نیاز دارد، پیش از حذف مدخل بازپخش آن محاسبه و ذخیره می‌شود، و پس از آن نقطه هیچ دسترسی تازه‌ای به داده‌ی هدف رخ نمی‌دهد. در پیاده‌سازی، این قید با ادعاهای اجرایی در ابتدا و انتهای پنجره‌ی دسترسی تضمین می‌شود، به‌گونه‌ای که نقض آن اجرا را متوقف می‌کند و نه اینکه بی‌صدا از آن بگذرد.

هم‌زمان باید تکرار کرد که چه چیزی حذف نمی‌شود: مدخل بازپخش وظیفه‌ی هدف از حافظه‌ی درون‌فرآیندی پاک می‌شود، اما داده‌ی خام روی دیسک، نقاط‌بازرسی، پشتیبان‌ها و نسخه‌های بیرونی موضوع این پروتکل نیستند. بودجه‌ی بازپخش هم بازتوزیع نمی‌شود؛ سهم آزادشده‌ی وظیفه‌ی حذف‌شده به وظایف باقی‌مانده داده نمی‌شود تا بهبود ظاهری ناشی از بزرگ‌ترشدن حافظه با اثر سازوکار فراموشی مخدوش نشود.

۳-۷ دیدگاه یکپارچه و جمع‌بندی فصل

دو مسیر این فصل، دو عملیاتی‌سازی از یک ایده‌ی واحدند: *ناحیه‌ی مشترک را بشناس، درون آن مختصه‌های حساس به نگه‌داری را جدا کن، و تنها همان‌ها را مهار کن.* تفاوت‌ها در سه محور است. محور اول، تعریف ناحیه‌ی مشترک: در EPALL این ناحیه اشتراک دقیق دو ماسک دودویی ذخیره‌شده است، در حالی که در PALL-Adapter از اشتراک دو مجموعه‌ی رتبه‌بندی‌شده‌ی گرادیانی می‌آید و بنابراین به آستانه‌ها حساس است. محور دوم، شکل مهار: EPALL یک جمله‌ی جریمه‌ی ℓ_2 در تابع هدف ترمیم به‌کار می‌برد که در طول کل حلقه‌ی ترمیم فعال است، در حالی که PALL-Adapter یک ضریب مقیاس روی گام به‌کار می‌برد که تنها در یک زیرفاز فعال است. محور سوم، دامنه‌ی پارامتری: EPALL روی کل ظرفیت شبکه عمل می‌کند و PALL-Adapter روی ماژول‌های کوچک روی یک بدنه‌ی منجمد.

از دید نتیجه، همان‌طور که فصل بعد نشان می‌دهد، محور اول و دوم به‌طور تعیین‌کننده مهم‌اند: مهار پایدار روی یک ناحیه‌ی دقیقا تعریف‌شده کار می‌کند و مقیاس‌دهی موقت روی یک ناحیه‌ی تقریبی، در نقطه‌ی پایان اثری نمی‌گذارد. این نتیجه، EPALL را روش اصلی این پایان‌نامه و PALL-Adapter را یک گزینه‌ی اکتشافی ممیزی‌شده می‌کند.

فصل ۴

آزمایش‌ها و نتایج

این فصل روش‌های فصل ۳ را به‌طور تجربی ارزیابی می‌کند. ترتیب ارائه‌ی نتایج از پرسش‌های پژوهش بخش ۱-۴ پیروی می‌کند: نخست مقایسه‌ی اصلی هم‌تراز و تحلیل جفتی آن (RQ1 و RQ2)، سپس محک موقعیت درخواست (RQ3)، آنگاه کنترل‌های مؤلفه‌ای هر دو مسیر (RQ4 و RQ5)، و در پایان حسابداری هزینه، ممیزی حریم خصوصی و مطالعات حساسیت.

یک قاعده‌ی گزارش‌دهی در سراسر فصل رعایت می‌شود: هر جدول تعداد بذرهای خود را مشخص می‌کند، هیچ سلولی رتبه‌بندی نمی‌شود مگر آنکه معیار به‌تنهایی قابل خوانش باشد، و نتایج با کمتر از پنج بذر صریحا *توصیفی* برچسب می‌خورند. مسیر بازتولید خروجی‌های تجمیعی و برچسب‌های آزمایش در پیوست مستند شده است.

۴-۱ طراحی آزمایش

۴-۱-۱ محک‌ها و افراز وظیفه‌ها

سه محک عمومی به‌کار می‌رود و افراز آن‌ها به وظیفه‌ها به‌گونه‌ای است که تعداد وظیفه و دشواری بازنمایی به‌تدریج افزایش یابد:

- CIFAR-10 [۷۴] به‌صورت ۵ وظیفه‌ی دوکلاسه؛ سطح تصادفی ۰/۵.
- CIFAR-100 [۷۴] در پروتکل هم‌پوشانی‌سنگین به‌صورت ۱۰ وظیفه‌ی پنج‌کلاسه (سطح تصادفی ۰/۲) و در پروتکل استاندارد به‌صورت وظیفه‌های ده‌کلاسه (سطح تصادفی ۰/۱).

- TinyImageNet [۷۵] به‌صورت ۲۰ وظیفه‌ی ده‌کلاسه؛ سطح تصادفی ۰/۱.

۴-۱-۲ سه پروتکل مجزا

نکته‌ی روش‌شناختی مهم این است که آزمایش‌ها در سه پروتکل *مجزا* اجرا شده‌اند و نتایج آن‌ها با یکدیگر مخلوط نمی‌شوند. تنها استثنا جاروب پراکندگی بخش ۴-۵ است که افراز و دنباله‌ی پروتکل استاندارد را با بودجه‌ی ۳ ایپاک ترکیب می‌کند؛ همان‌جا تصریح شده که اعداد آن تنها درون خود جدول قابل مقایسه‌اند. سه پروتکل چنین‌اند:

پروتکل استاندارد. ۲۰ ایپاک به‌ازای هر وظیفه، بدنه‌ی آموزش‌دیده از صفر، هشت بذر $\{۰, \dots, ۷\}$. این پروتکل، مبنای مقایسه‌ی اصلی بخش ۴-۳ است.

پروتکل هم‌پوشانی‌سنگین. ۳ ایپاک به‌ازای هر وظیفه و افراز پنج‌کلاسه‌ی CIFAR-100. این پروتکل عمدا هم‌پوشانی بیشتری بین زیرشبکه‌ها ایجاد می‌کند و برای کنترل‌های مؤلفه‌ای EPALL و مقایسه‌ی سه نسخه‌ی PALL به‌کار می‌رود.

پروتکل پیش‌آموزش‌دیده. بدنه‌ی ImageNet [۷۶] منجمد. این پروتکل، فرض طبیعی روش‌های کارآمد پارامتری را برقرار می‌کند و برای مقایسه‌ی LoRA و PALL-Adapter استفاده می‌شود.

در هر سه پروتکل، دنباله‌ی درخواست‌ها هر وظیفه را حداکثر یک‌بار آموزش می‌دهد، سه درخواست فراموشی دارد، و هیچ وظیفه‌ی حذف‌شده‌ای را دوباره معرفی نمی‌کند. درون هر بذر، همه‌ی روش‌ها *دنباله‌ی درخواست یکسانی* می‌بینند؛ همین جفت‌شدگی است که تحلیل جفتی بخش ۴-۳-۲ را ممکن می‌کند.

۴-۱-۳ معماری‌ها و ابرپارامترها

به‌جز آزمون ViT، اجراهای CIFAR-10، TinyImageNet و پیش‌آموزش‌دیده از نسخه‌های ResNet-18 و اجراهای اصلی CIFAR-100 از نسخه‌های ResNet-34 استفاده می‌کنند [۲۹]. بهینه‌ساز SGD [۲۵] با نرخ یادگیری ۰/۰۱، تکانه‌ی ۰/۹، تنظیم وزن ۵×۱۰^{-۴} و اندازه‌ی دسته‌ی ۳۲ است. بودجه‌ی بازپخش ۵۰۰ نمونه و از پیش میان وظایف افراز می‌شود، بدون بازتوزیع پس از حذف. تنها استثنا، LoRA روی Split-CIFAR-100 است که نرخ یادگیری $۱۰^{-۳}$ می‌گیرد، چون با $۱۰^{-۲}$ واگرا می‌شود.

پیکربندی روش‌های پیشنهادی چنین است. EPALL: نسبت محافظت $\rho = ۰/۲$ و ضریب لنگر $\lambda = ۱$. PALL-Adapter: گلوگاه وظیفه و گلوگاه مشترک ۱۶، $\alpha_f = ۰/۳$، $\alpha_p = ۰/۲$، $p = |S_{\text{crit}}|/|S_{\text{forget}}|$،

ده گام فراموشی و پنجاه گام ترمیم. رتبه‌ی LoRA برابر ۸. پراکندگی زیرشبکه ۰/۸ روی CIFAR-10 و TinyImageNet و ۰/۹ روی CIFAR-100 است. پیکربندی کامل آزمایش‌ها در جدول ۵-۱ در پیوست آمده است.

۴-۱-۴ بذرها و سخت‌افزار

مقایسه‌ی اصلی استاندارد، کنترل‌های مستقیم EPALL و تحلیل لنگر از بذرهای $\{۰,\dots,۷\}$ استفاده می‌کنند؛ جدول‌های پیش‌آموزش‌دیده، هم‌پوشانی‌سنگین و مؤلفه‌های آداپتر از $\{۰,۱,۲\}$؛ و تحلیل‌های معیار اهمیت، ViT و ماندگاری از $\{۰,۱\}$. محک موقعیت درخواست و جاروب پراکندگی هر دو پنج بذر $\{۰,\dots,۴\}$ دارند و جاروب شدت ماسک پیوسته سه بذر. مرجع بازآموزی دو بذر، MIA اصلاح‌شده سه بذر و حسابداری حافظه یک دنباله‌ی هم‌تراز دارد. جاروب عرض گلوگاه نیز دو بذر دارد و صرفا برای نشان‌دادن غیریکنوایی به‌کار می‌رود. پیکربندی سخت‌افزاری و نرم‌افزاری اجراها، به‌همراه نسخه‌ی مخزن و درهم‌سازی دنباله‌های درخواست، نیز در پیوست آمده است.

۴-۲ معیارهای ارزیابی

پنج معیار گزارش می‌شود و تفسیر هر یک به‌دقت تعریف می‌شود.

دقت نهایی نگه‌داری A_{final} (بیشتر بهتر). میانگین دقت روی وظایف فعال در پایان دنباله.

فراموشی میانگین F_{avg} (کمتر بهتر). برای درخواست r:

$$F_{\text{avg}}(r) = \frac{۱}{|\mathcal{T}_{\text{trained}}|} \sum_{t} \Big[\max_{k<r} A_t(k) - A_t(r) \Big]. \tag{۱-۴}$$

این معیار روی همه‌ی وظایفی محاسبه می‌شود که تا لحظه‌ی r آموزش دیده‌اند، از جمله وظایفی که بعدا حذف شده‌اند. به همین دلیل F_{avg} وضعیت *کل* مسیر یادگیری پیوسته را می‌سنجد و نه اثر یک درخواست مشخص را؛ برای سنجش اثر یک درخواست از دو معیار بعدی استفاده می‌شود.

افت نگه‌داری F_u و بیشترین افت (کمتر بهتر). F_u اختلاف میانگین دقت وظایف نگه‌داری‌شده پیش و پس از یک درخواست است، و

$$\text{WorstDrop} = \max_{t \in \mathcal{T}_a} \big(A_t^{\text{before}} - A_t^{\text{after}} \big). \tag{۲-۴}$$

این معیار *علامت‌دار* گزارش می‌شود و در صفر بریده نمی‌شود؛ بنابراین مقدار منفی به این معنا است که ترمیم، بدترین وظیفه را بهبود داده است و این اطلاعات با بریدن از دست می‌رفت.

دقت وظیفه‌ی فراموش‌شده A_u. بی‌درنگ پس از درخواست اندازه‌گیری می‌شود و همیشه از طریق $|A_u - c|$ خوانده می‌شود که c سطح تصادفی همان وظیفه است. تأکید می‌شود که این معیار «کمتر بهتر» نیست.

زمان درخواست T_f و نسبت دامنه‌ی ترمیم R_{upd}. T_f زمان دیواری پردازش درخواست است و با `time.perf_counter` اندازه‌گیری می‌شود. چون فراخوانی‌های CUDA ناهم‌گام‌اند و پیش از خواندن ساعت هیچ `torch.cuda.synchronize` اجرا نمی‌شود، این عدد شامل هر کاری است که تا لحظه‌ی خواندن ساعت روی دستگاه هم‌گام شده باشد و نه لزوما تمام کار صف‌شده. بنابراین T_f یک *تشخیص* مرتبه‌ای است و نه یک اندازه‌گیری دقیق زمان محاسبه؛ همه‌ی نتیجه‌گیری‌های این پایان‌نامه درباره‌ی زمان، در همین سطح مرتبه‌ای بیان شده‌اند. R_{upd} *تنها* نسبت اسمی پارامترهای آموزش‌پذیر در ماسک ترمیم نگه‌داری است؛ این معیار محاسبات کل، حافظه، زمان اجرا یا تعداد مختصه‌های واقعا تغییریافته را نمی‌سنجد. بخش ۴-۱۰ به همین دلیل، حالت مقیم را جداگانه حسابداری می‌کند.

جدول‌ها از آخرین رویداد فراموشی هر اجرا استفاده می‌کنند و سپس میانگین و انحراف معیار نمونه‌ای را روی بذرها می‌گیرند.

۴-۳ مقایسه‌ی اصلی استاندارد

جدول‌های ۴-۱ و ۴-۲ مقایسه‌ی اصلی این پایان‌نامه‌اند. یازده روش، هشت بذر، و درون هر بذر یک دنباله‌ی درخواست مشترک. سطرها بر اساس *فرض عملیاتی* گروه‌بندی شده‌اند و نه بر اساس عملکرد، چون مقایسه‌ی مستقیم روشی که یک شبکه به‌ازای هر وظیفه نگه می‌دارد با روشی که یک مدل مشترک دارد، بی‌معنا است. SSD و SalUn از تجمیع هم‌پروتکل خود می‌آیند؛ اجراهای تک‌بذری تنظیم SSD در این مقایسه وارد نشده‌اند.

جدول ۴-۱: مقایسه‌ی هم‌تراز روی Split-CIFAR-10 استاندارد؛ ۲۰ ایپاک به‌ازای هر وظیفه، هشت بذر، و درون هر بذر دنباله‌ی درخواست مشترک. سطرها بر اساس فرض عملیاتی گروه‌بندی شده‌اند و نام روش‌های این پایان‌نامه پرنگ است. هیچ سلول عددی رتبه‌بندی نشده، چون معیارها باید هم‌زمان خوانده شوند. سطح تصادفی دقت هدف ۰/۵ است. †سطرهای PEFT در این پروتکل روی بدنه‌ی منجمد با مقدارگذاری تصادفی می‌نشینند؛ رژیم موردنظر آن‌ها در جدول ۴-۱۰ است.

خانواده	روش	دقت نهایی ↑	فراموشی ↓	بیشترین افت ↓	دقت هدف ۰/۵ →	زمان (ث) ↓
جداسازی	CLPU	0.9500 ± 0.0169	0.2803 ± 0.0266	0.0000 ± 0.0000	0.5041 ± 0.0152	0.0005
یادگیری پیوسته	DER++	0.8891 ± 0.0449	0.2773 ± 0.0535	0.0002 ± 0.0113	0.5937 ± 0.1343	2.8334
	ER	0.8679 ± 0.0464	0.3022 ± 0.0471	0.0253 ± 0.0317	0.4918 ± 0.0521	2.2502
	EWC	0.7258 ± 0.1421	0.2912 ± 0.0983	0.0000 ± 0.0000	0.8122 ± 0.2102	0.0001
	LwF	0.9168 ± 0.0317	0.0781 ± 0.0448	0.0000 ± 0.0000	0.9469 ± 0.0280	0.0001
یادگیری‌زدایی	SSD	0.8639 ± 0.0614	0.1814 ± 0.0938	0.0126 ± 0.0148	0.8453 ± 0.1589	0.2676
	SalUn	0.8555 ± 0.0473	0.3168 ± 0.0527	0.0291 ± 0.0443	0.5039 ± 0.0986	2.3862
PALL کامل	PALL	0.9346 ± 0.0218	0.2855 ± 0.0253	0.0119 ± 0.0104	0.4960 ± 0.0278	2.7369
	EPALL	0.9433 ± 0.0160	0.2872 ± 0.0249	0.0027 ± 0.0031	0.4863 ± 0.0216	4.8349
†PEFT	LoRA†	0.7285 ± 0.0730	0.1539 ± 0.0457	0.0219 ± 0.0216	0.4936 ± 0.0132	3.4968
	PALL-Adapter†	0.7222 ± 0.0715	0.1556 ± 0.0428	0.0117 ± 0.0162	0.5097 ± 0.0349	3.8352

جدول ۴-۲: مقایسه‌ی هم‌تراز روی Split-CIFAR-100 استاندارد؛ ۲۰ ایپاک به‌ازای هر وظیفه، هشت بذر، و درون هر بذر دنباله‌ی درخواست مشترک. سطح تصادفی دقت هدف ۰/۱ است. سایر نمادها مانند جدول ۴-۱.

خانواده	روش	دقت نهایی ↑	فراموشی ↓	بیشترین افت ↓	دقت هدف ۰/۱ →	زمان (ث) ↓
جداسازی	CLPU	0.7355 ± 0.0069	0.1935 ± 0.0151	0.0000 ± 0.0000	0.1047 ± 0.0107	0.0007
یادگیری پیوسته	DER++	0.6532 ± 0.0308	0.2459 ± 0.0209	0.0341 ± 0.0145	0.3443 ± 0.1913	12.3429
	ER	0.5923 ± 0.0389	0.2797 ± 0.0277	0.0515 ± 0.0309	0.3956 ± 0.1079	6.9605
	EWC	0.4359 ± 0.0522	0.3939 ± 0.0247	0.0000 ± 0.0000	0.4954 ± 0.2044	0.0001
	LwF	0.6661 ± 0.0333	0.1540 ± 0.0152	0.0000 ± 0.0000	0.7132 ± 0.1180	0.0001
یادگیری‌زدایی	SSD	0.4529 ± 0.2101	0.4340 ± 0.1566	0.0295 ± 0.0365	0.2619 ± 0.0801	0.5007
	SalUn	0.5892 ± 0.0480	0.2869 ± 0.0270	0.0583 ± 0.0407	0.3967 ± 0.1222	6.9425
PALL کامل	PALL	0.7223 ± 0.0168	0.2068 ± 0.0144	0.0222 ± 0.0138	0.1023 ± 0.0040	12.6364
	EPALL	0.7371 ± 0.0093	0.1958 ± 0.0121	0.0056 ± 0.0062	0.0959 ± 0.0075	24.6999
†PEFT	LoRA†	0.2695 ± 0.0163	0.0652 ± 0.0128	0.0310 ± 0.0174	0.1042 ± 0.0095	7.5442
	PALL-Adapter†	0.2583 ± 0.0208	0.0755 ± 0.0080	0.0373 ± 0.0227	0.1061 ± 0.0072	8.8310

۴-۳-۱ خوانش نتایج اصلی

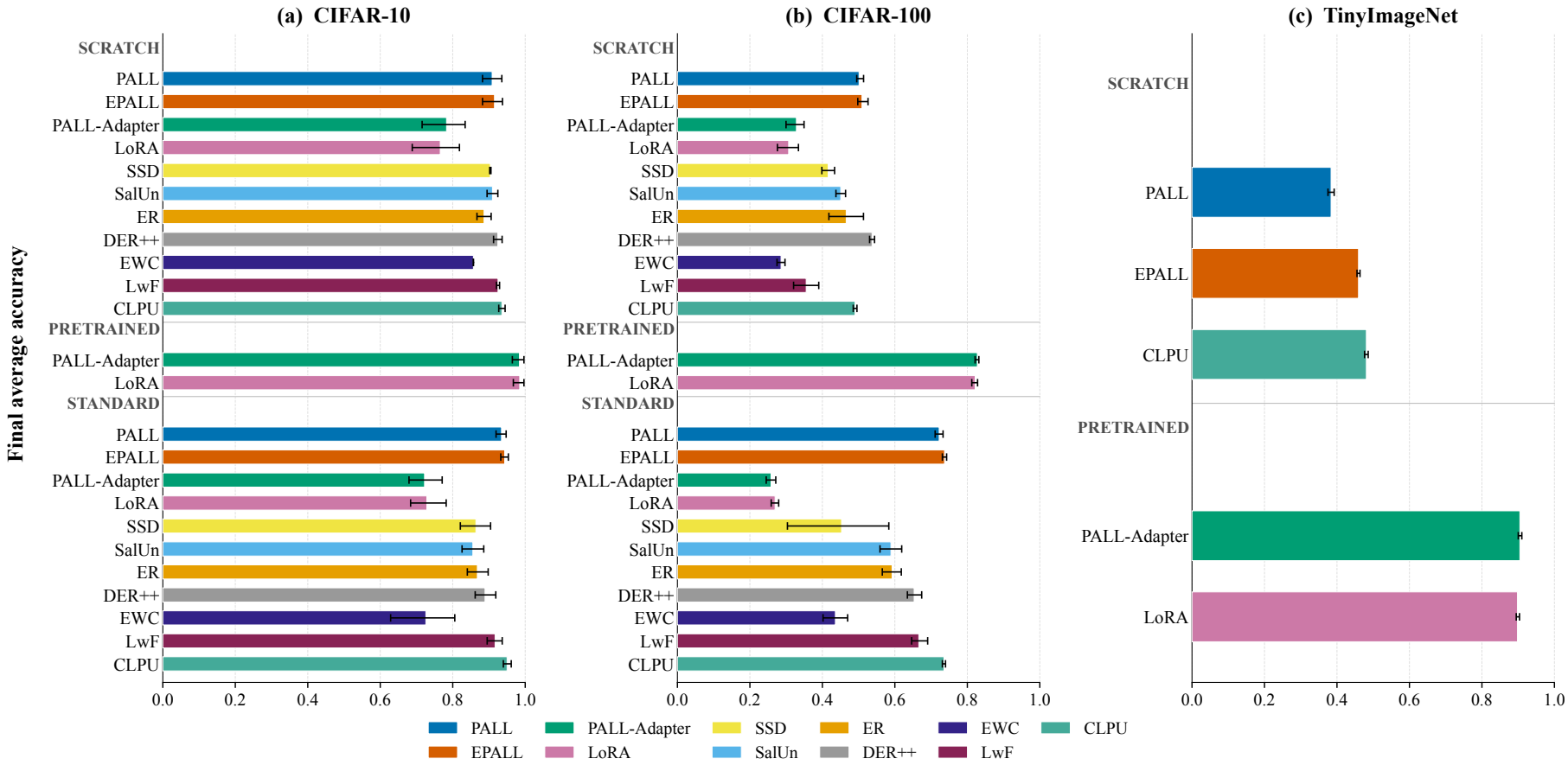
CLPU قوی‌ترین سطر است و ما ادعای برتری بر آن نداریم. پیاده‌سازی ارزیابی‌شده یک شبکه‌ی کامل به‌ازای هر وظیفه‌ی فعال نگه می‌دارد و فراموشی را با پاک‌کردن همان مدخل انجام می‌دهد. بنابراین زمان درخواست زیرمیلی‌ثانیه‌ای آن، هزینه‌ی یک حذف از دیکشنری را می‌سنجد و نه هزینه‌ی نگه‌داشتن آن شبکه‌ها را. CLPU وقتی چنین جداسازی‌ای قابل قبول باشد گزینه‌ی جذابی است، اما یک راه‌حل مدل‌مشترک نیست. بخش ۴-۱۰ هزینه‌ی پنهان آن را آشکار می‌کند.

میان روش‌های مدل‌مشترک، EPALL به مرجع جداسازی نزدیک می‌شود. روی CIFAR-10 فاصله‌ی A_{final} آن از CLPU کمتر از یک واحد درصد است (۰/۹۴۳۳ در برابر ۰/۹۵۰۰) و روی CIFAR-100 دو مقدار عددا نزدیک‌اند (۰/۷۳۷۱ در برابر ۰/۷۳۵۵). این نزدیکی نه برتری است و نه هم‌ارزی اثبات‌شده: اختلاف نسبت به پراکندگی بذرها کوچک است و هیچ آزمون هم‌ارزی برای آن اجرا نشده است. هم‌زمان A_u را به سطح تصادفی می‌رساند (۰/۴۸۶۳ و ۰/۰۹۵۹) و بیشترین افت را در ۰/۰۰۲۷ و ۰/۰۰۵۶ نگه می‌دارد. بنابراین بدون دادن یک شبکه‌ی کامل به هر وظیفه، عملا به همان نقطه می‌رسد.

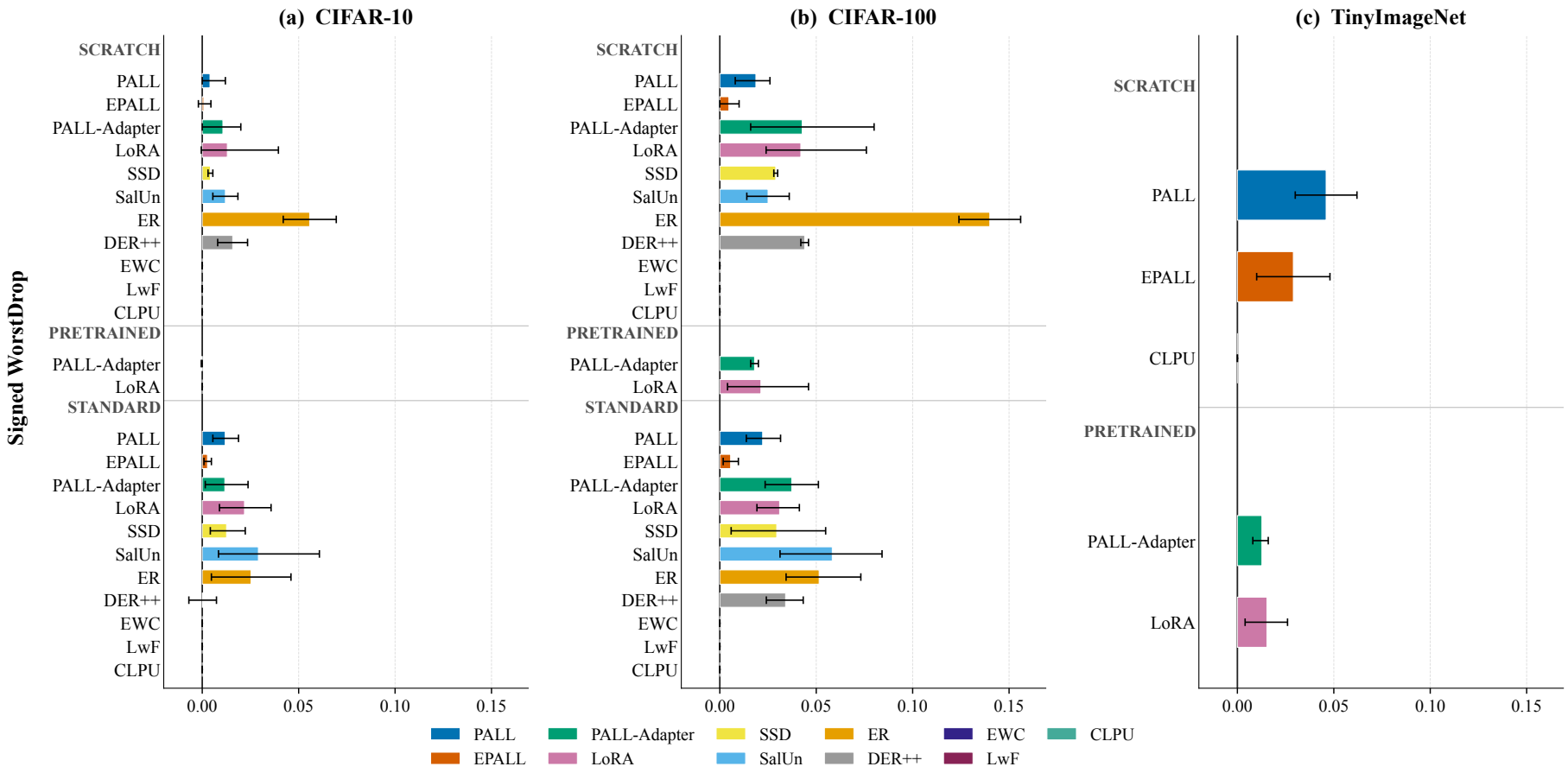
ستون A_u روش‌های حفظ‌محور را رد می‌کند. EWC و LwF روی CIFAR-10 بیشترین افت صفر ثبت می‌کنند — که در نگاه اول عالی است — اما A_u آن‌ها ۰/۸۱۲۲ و ۰/۹۴۶۹ است؛ یعنی وظیفه‌ی هدف عملا حذف نشده و بخش عمده‌ی رفتار آن حفظ شده است. به همین ترتیب، SSD روی CIFAR-10 با $A_u =$ ۰/۸۴۵۳ می‌ماند و هر دو روش SSD و SalUn روی CIFAR-100 با ۰/۲۶۱۹ و ۰/۳۹۶۷ به‌روشنی بالای هدف ۰/۱ باقی می‌مانند. این دقیقا نمونه‌ای است که چرا معیارها باید هم‌زمان خوانده شوند: هیچ‌یک از این دو ستون به‌تنهایی داوری درستی نمی‌دهد.

دو سطر PEFT تشخیصی‌اند و رقیب هم‌ظرفیت نیستند. در این پروتکل از صفر، هم LoRA و هم PALL-Adapter روی یک ResNet منجمد با مقدارگذاری تصادفی نشسته‌اند. دقت CIFAR-100 آن‌ها به‌طور مشابه پایین است (۰/۲۶۹۵ و ۰/۲۵۸۳) به‌رغم آنکه سازوکار فراموشی‌شان متفاوت است، و هر دو با پیش‌آموزش‌دیده‌شدن بدنه بازیابی می‌شوند (جدول ۴-۱۰). این الگو به یک گلوگاه *بازنمایی* اشاره می‌کند و نه به گام فراموشی.

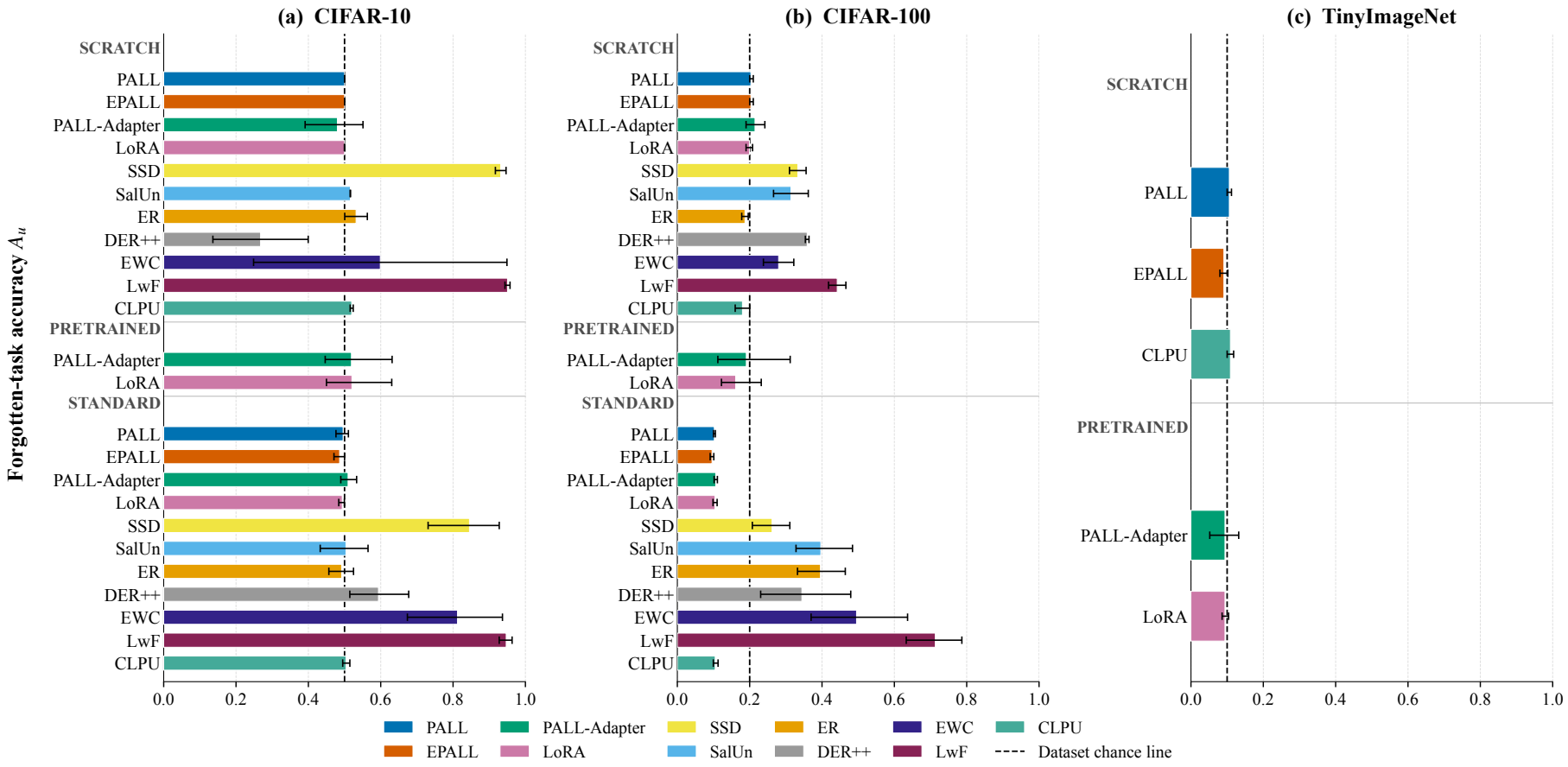
شکل ۴-۱ دقت نهایی نگه‌داری، شکل ۴-۲ بیشترین افت و شکل ۴-۳ دقت وظیفه‌ی فراموش‌شده را برای هر سه مجموعه‌داده و هر رژیم موجود نشان می‌دهند. جداسازی هر معیار در یک صفحه‌ی افقی، خوانایی برچسب‌ها و بازه‌های اطمینان را در نسخه‌ی چاپی حفظ می‌کند.



شکل ۴-۱: دقت نهایی به تفکیک مجموعه‌داده و رژیم. رژیم‌ها تجمیع نشده‌اند؛ میله‌ها میانگین جدول و خطاها بازه‌ی بوت‌استرپ ۹۵ درصد بذرهای اجراهای کامل‌شده هستند.



شکل ۴-۲: بیشترین افت علامت‌دار به تفکیک مجموعه‌داده و رژیم. رژیم‌ها تجمیع نشده‌اند؛ خط عمودی صفر حفظ شده و مقادیر منفی، بهبود پس از ترمیم را نشان می‌دهند.



شکل ۴-۳: A_u به تفکیک مجموعه‌داده و رژیم. رژیم‌ها تجمیع نشده‌اند؛ خط‌چین سطح تصادفی هر مجموعه‌داده را نشان می‌دهد.

۴-۳-۲ تحلیل جفتی و آزمون معناداری

چون دنباله‌ها بر اساس بذر جفت شده‌اند، مقایسه‌ی مستقیم EPALL با PALL اصلی ممکن است. جدول ۴-۳ اختلاف‌های جفتی و بازه‌های بوت‌استرپ توصیفی را نشان می‌دهد.

جدول ۴-۳: اختلاف‌های جفتی (EPALL منهای PALL) روی بذرهای یکسان. مقدار مثبت برای دقت نهایی، فراموشی و بیشترین افت به سود EPALL است؛ ΔT_f منفی یعنی EPALL کندتر است. بازه‌های بوت‌استرپ روی هشت بذر هم‌تراز و توصیفی‌اند.

مجموعه‌داده	جفت	Δ دقت نهایی	Δ فراموشی	Δ بیشترین افت	ΔT_f (ث)
CIFAR-10	8	0.0087 [0.0034, 0.0151]	−0.0017 [−0.0078, 0.0033]	0.0092 [0.0029, 0.0157]	−2.10
CIFAR-100	8	0.0147 [0.0062, 0.0229]	0.0110 [0.0055, 0.0160]	0.0166 [0.0094, 0.0236]	−12.06

روی CIFAR-100 اختلاف A_{final} مثبت و برابر ۰/۰۱۴۷+ با بازه‌ی [۰/۰۰۶۲, ۰/۰۲۲۹] است، با بهبود میانگین ۰/۰۱۱۰ در F_{avg} و ۰/۰۱۶۶ در بیشترین افت؛ هیچ‌یک از این سه بازه صفر را در بر نمی‌گیرد. روی CIFAR-10 افزایش دقت ۰/۰۰۸۷+ [۰/۰۰۳۴, ۰/۰۱۵۱] و بهبود بیشترین افت ۰/۰۰۹۲ [۰/۰۰۲۹, ۰/۰۱۵۷] است، در حالی که F_{avg} عملا مختلط است (۰/۰۰۱۷− با بازه‌ای که صفر را در بر می‌گیرد). هزینه‌ی این نگه‌داری، زمان درخواست است: EPALL حدود ۲/۱۰ ثانیه روی CIFAR-10 و ۱۲/۰۶ ثانیه روی CIFAR-100 از PALL اصلی کندتر است.

جدول ۴-۴ آزمون‌های دقیق یک‌طرفه را گزارش می‌کند. چون شش آزمون هم‌زمان انجام می‌شود، مقدار p خام به‌تنهایی کافی نیست و اصلاح Holm برای چندگانگی اعمال شده است. روی CIFAR-100 هر سه معیار به $p =$ ۰/۰۰۷۸۱۲ می‌رسند و هفت جفت از هشت جفت به سود EPALL است، با اندازه‌های اثر جفتی d_z برابر ۱/۱۱، ۱/۵۱ و ۱/۳۵؛ هر سه پس از اصلاح Holm نیز در سطح ۰/۰۵ باقی می‌مانند. روی CIFAR-10 تنها A_{final} باقی می‌ماند ($p =$ ۰/۰۰۷۸۱۲ و پس از Holm برابر ۰/۰۴۶۹).

جدول ۴-۴: آزمون‌های دقیق یک‌طرفه‌ی جفتی (EPALL در برابر PALL) روی بذرهای یکسان. چون شش آزمون هم‌زمان انجام می‌شود، ستون Holm مقدار p اصلاح‌شده برای چندگانگی را می‌دهد و ستون آخر نشان می‌دهد کدام آزمون در سطح ۰/۰۵ باقی می‌ماند. کف قابل‌دستیابی p برای هر سطر جداگانه است و *یکسان نیست*: برای سطرهای CIFAR-100 برابر ۰/۰۰۳۹۰۶ و برای دقت نهایی CIFAR-10 برابر ۰/۰۰۷۸۱۲ است. برای معیارهای «کمتر بهتر» علامت اختلاف وارونه شده تا مقدار مثبت همیشه به سود EPALL باشد.

داده	معیار	موافق/گره	میانگین Δ	d_z	ویلکاکسون	Holm	کف p	بقا
CIFAR-10	دقت نهایی	7 / 1	0.008656	0.9561	0.007812	0.046872	0.007812	**بله**
CIFAR-10	بیشترین افت	5 / 2	0.009187	0.9308	0.03125	0.0625	0.015625	خیر
CIFAR-10	فراموشی	3 / 1	−0.00170	−0.1950	0.59375	0.59375	0.007812	خیر
CIFAR-100	دقت نهایی	7 / 0	0.014732	1.1141	0.007812	0.046872	0.003906	**بله**
CIFAR-100	بیشترین افت	7 / 0	0.016625	1.5106	0.007812	0.046872	0.003906	**بله**
CIFAR-100	فراموشی	7 / 0	0.011049	1.3487	0.007812	0.046872	0.003906	**بله**

تفسیر این جدول باید محتاطانه بماند و دو تصحیح مهم دارد. نخست، **بیشترین افت روی CIFAR-10 پس از اصلاح Holm معنادار نیست**: مقدار خام ۰/۰۳۱۲۵ پس از اصلاح به ۰/۰۶۲۵ می‌رسد و از سطح ۰/۰۵ می‌گذرد. آزمون علامت همان معیار نیز به $p =$ ۰/۱۰۹ می‌رسد، پس شاهد آن از دو جهت ضعیف است و در این پایان‌نامه به‌عنوان یافته‌ی معنادار گزارش نمی‌شود. دوم، **کف p قابل‌دستیابی برای همه‌ی سطرها یکسان نیست**: این کف به تعداد جفت‌های بی‌گره بستگی دارد و برای سطرهای CIFAR-100 (بدون گره) برابر ۰/۰۰۳۹۰۶ است، نه ۰/۰۰۷۸۱۲. بنابراین مقادیر مشاهده‌شده‌ی CIFAR-100 *روی کف نایستاده‌اند* بلکه دو برابر آن هستند، و این یعنی توان آزمون کاملا اشباع نشده است. تنها معیاری که در هیچ مجموعه‌داده‌ای بهبود پیوسته نشان نمی‌دهد F_{avg} روی CIFAR-10 است، که با علامت منفی اختلاف جفتی در جدول ۴-۳ سازگار است.

جمع‌بندی محافظه‌کارانه: از شش آزمون، **چهار آزمون** پس از اصلاح چندگانگی باقی می‌مانند — هر سه معیار روی CIFAR-100 و دقت نهایی روی CIFAR-10.

یادداشت درباره‌ی تعداد بذر. نتایج این بخش بر هشت بذر (۰ تا ۷) استوارند که وضعیت کامل‌شده‌ی مخزن نتایج است. نسخه‌ی مقاله‌ی کنفرانسی همین کار بر پنج بذر (۰ تا ۴) بسته شده بود و اعداد آن با

اعداد این‌جا تفاوت جزئی دارد؛ جهت همه‌ی نتیجه‌گیری‌ها یکسان مانده است، اما مقادیر عددی و pها متفاوت‌اند. هرجا در این پایان‌نامه عددی گزارش می‌شود، منبع آن وضعیت کنونی مخزن نتایج است و نه مقاله.

۴-۴ محک موقعیت درخواست

ادعای انگیزشی EPALL این بود که محافظت *با رشد هم‌پوشانی* اهمیت بیشتری می‌یابد. برای آزمون این ادعا یک محک با پنج درجه‌ی از پیش تعیین‌شده ساخته شد: پنج درجه × شش روش × دو مجموعه‌داده × پنج بذر ۳۰۰ = اجرای هم‌تراز.

اما باید تصریح شود که این محک، هم‌پوشانی را کنترل نمی‌کند. آنچه درجه‌ها واقعا تغییر می‌دهند، *موقعیت درخواست فراموشی در دنباله* است؛ یعنی تعداد وظایفی که پس از وظیفه‌ی هدف آموزش می‌بینند. این کمیت را K می‌نامیم: روی CIFAR-10 برابر $K \in \{۰, ۱, ۲, ۳, ۴\}$ و روی CIFAR-100 برابر $K \in \{۰, ۲, ۴, ۷, ۹\}$ است. چون افراز کلاس‌ها و شناسه‌های وظیفه در این خط لوله ثابت‌اند، کنترل هم‌پوشانی *واقعی* ماسک‌ها به تغییر خود افراز داده نیاز دارد که در این کار انجام نشده است.

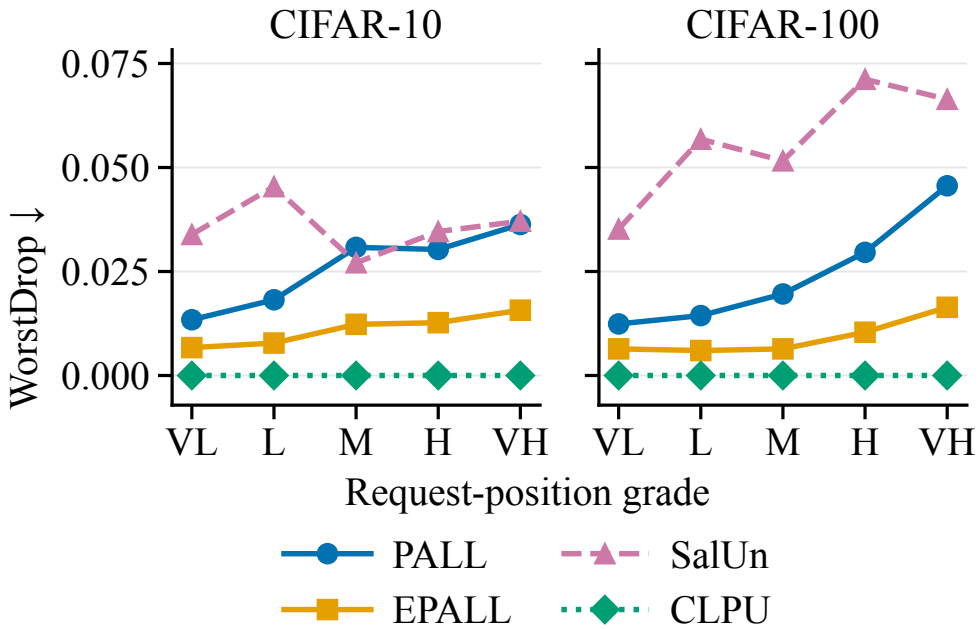
در نتیجه K تنها یک *شاخص جانشین* برای هم‌پوشانی مورد انتظار است و با تغییر آن، چند عامل هم‌زمان جابه‌جا می‌شوند: هویت وظیفه‌ی هدف، سن آن وظیفه در لحظه‌ی حذف، ترتیب درخواست‌ها، و مقدار آموزشی که پس از هدف انجام می‌شود. بنابراین این محک را در سراسر پایان‌نامه **محک موقعیت درخواست** می‌نامیم و نتیجه‌ی آن را یک *روند توصیفی سازگار با فرضیه* می‌خوانیم، نه شاهد علّی. جدول ۴-۵ بیشترین افت را در همه‌ی این خانه‌ها گزارش می‌کند.

جدول ۴-۵: بیشترین افتِ علامت‌دار در پنج درجه‌ی از پیش تعیین‌شده‌ی *موقعیت درخواست*، از بسیار کم تا بسیار زیاد؛ پنج بذر در هر خانه، شش روش و دو مجموعه‌داده، در مجموع ۳۰۰ اجرای هم‌تراز (۵ × ۲ × ۶ × ۵). درجه‌ها تعداد وظایف آموزش‌دیده پس از وظیفه‌ی هدف را تغییر می‌دهند و هم‌پوشانی واقعی ماسک‌ها را کنترل نمی‌کنند؛ هم‌زمان با آن، هویت و سن وظیفه‌ی هدف نیز تغییر می‌کند. مقادیر LoRA و PALL-Adapter بازتاب کف ظرفیت روی بدنه‌ی منجمد تصادفی‌اند؛ تنها برای کامل‌بودن آورده شده‌اند.

مجموعه‌داده	روش	بسیار کم	کم	متوسط	زیاد	بسیار زیاد
CIFAR-10	CLPU	0.0000	0.0000	0.0000	0.0000	0.0000
	PALL	0.0134	0.0182	0.0308	0.0303	0.0363
	EPALL	0.0067	0.0078	0.0123	0.0127	0.0157
	SalUn	0.0339	0.0454	0.0271	0.0346	0.0371
	LoRA	-0.0075	0.0041	-0.0160	0.0049	0.0049
	PALL-Adapter	0.0054	0.0071	-0.0024	0.0002	0.0036
CIFAR-100	CLPU	0.0000	0.0000	0.0000	0.0000	0.0000
	PALL	0.0124	0.0144	0.0196	0.0296	0.0456
	EPALL	0.0064	0.0060	0.0064	0.0104	0.0164
	SalUn	0.0352	0.0568	0.0516	0.0712	0.0664
	LoRA	0.0868	0.0828	0.0788	0.0780	0.0528
	PALL-Adapter	0.0620	0.0588	0.0584	0.0564	0.0672

الگوی CIFAR-100 همان چیزی است که سازوکار پیش‌بینی می‌کرد: بیشترین افت PALL اصلی با افزایش شاخص موقعیت درخواست از ۰/۰۱۲۴ به ۰/۰۴۵۶ رشد می‌کند، در حالی که EPALL در بازه‌ی ۰/۰۰۶۰ تا ۰/۰۱۶۴ نسبتا مسطح می‌ماند. روی CIFAR-10 همین ترتیب حفظ می‌شود اما در بازه‌ای کوچک‌تر. SalUn رفتاری مشابه PALL اصلی نشان می‌دهد و CLPU همان‌گونه که انتظار می‌رفت به‌طور ساختاری صفر می‌ماند، چون شبکه‌های مستقل امکان تداخل ندارند. شکل ۴-۴ همین روند را نمایش می‌دهد.

هشدار روش‌شناختی. این یک روند *درجه‌به‌درجه* و توصیفی است و هیچ ادعای علّی از آن بیرون نمی‌آید، چون درجه هم‌زمان چند عامل را جابه‌جا می‌کند. هم‌پوشانی *محقق‌شده* میان روش‌ها متفاوت است و رگرسیون‌های تجمیعی روی هم‌پوشانی اندازه‌گیری‌شده عمدتا صفر را در بر می‌گیرند. جدول ۴-۶ این رگرسیون‌ها را گزارش می‌کند: تنها استثنا شیب CIFAR-100 مسیر آداپتر است. توجه شود که



شکل ۴-۴: بیشترین افت در پنج درجه‌ی موقعیت درخواست برای چهار روش اصلیِ نمایش‌داده‌شده (میانگین پنج اجرای هم‌تراز هر درجه؛ کمتر بهتر). درجه‌های محور افقی از «بسیار کم» تا «بسیار زیاد»، تعداد وظایف آموزش‌دیده پس از وظیفه‌ی هدف را کدگذاری می‌کنند و شاخصی جانشین برای هم‌پوشانی مورد انتظارند، نه خود هم‌پوشانی. PALL اصلی با افزایش این شاخص بالا می‌رود، در حالی که EPALL نسبتا مسطح می‌ماند؛ CLPU به‌طور ساختاری صفر است. دو روش دیگر در جدول ۴-۵ آمده‌اند.

سنجه‌ی محور افقی برای دو خانواده یکسان نیست — برای PALL کامل، میانگین اشتراک‌ـبرـاجتماع ماسک‌های زیرشبکه و برای آداپتر، نسبت بحرانی مشترک — پس مقایسه‌ی بزرگی شیب‌ها میان این دو بازنمایی نیازمند احتیاط است.

جدول ۴-۶: رگرسیون بیشترین افتِ علامت‌دار روی هم‌پوشانی *اندازه‌گیری‌شده*. بازه‌ها از خطای استاندارد مقاوم HC3 با مقادیر بحرانی توزیع t می‌آیند و اجراهای تکراری به آخرین اجرای کامل خود تجمیع شده‌اند. سنجه‌ی محور افقی میان دو خانواده یکسان نیست، پس بزرگی شیب‌ها مستقیما قابل مقایسه نیست.

داده	روش	سنجه‌ی محور x	n	شیب	بازه‌ی ۹۵٪	شامل صفر
CIFAR-100	PALL	اشتراک‌براجتماع ماسک	11	0.5173	$[-1124.37, 1125.41]$	بله
CIFAR-100	EPALL	اشتراک‌براجتماع ماسک	12	0.0652	$[-0.2677, 0.3980]$	بله
CIFAR-100	PALL-Adapter	نسبت بحرانی مشترک	86	0.0388	$[0.0227, 0.0548]$	**خیر**
CIFAR-10	PALL	اشتراک‌براجتماع ماسک	12	-35.9451	$[-209.28, 137.39]$	بله
CIFAR-10	EPALL	اشتراک‌براجتماع ماسک	14	3.8640	$[-107.69, 115.42]$	بله
CIFAR-10	PALL-Adapter	نسبت بحرانی مشترک	92	0.0085	$[0.0003, 0.0166]$	**خیر**

بازه‌های بسیار پهن سطرهای PALL کامل، پیامد مستقیم آن است که هم‌پوشانی اندازه‌گیری‌شده در آن اجراها تنها سه یا چهار مقدار یگانه دارد؛ بنابراین این جدول را باید نشانه‌ی *کمبود تنوع در متغیر مستقل* خواند و نه شاهد نبود اثر. همین محدودیت، دلیل اصلی طراحی محک سطح‌محور بود.

۴-۵ جاروب پراکندگی و هم‌پوشانی اندازه‌گیری‌شده

محک بخش ۴-۴ یک ضعف روش‌شناختی دارد که در همان‌جا تصریح شد: درجه‌ها *موقعیت درخواست* را تغییر می‌دهند و هم‌پوشانی تنها به‌عنوان یک اثر جانبی جابه‌جا می‌شود؛ هم‌زمان هویت وظیفه‌ی هدف، سن آن و مقدار آموزش پس از آن نیز تغییر می‌کنند. بنابراین آن محک با فرضیه سازگار است اما آن را نمی‌آزماید. این بخش یک آزمون مکمل ارائه می‌کند که در آن موقعیت درخواست ثابت است و دستگیره‌ی آزمایشی، پراکندگی زیرشبکه است. چون این دستگیره ظرفیت را نیز تغییر می‌دهد، نباید آن را «دستکاری مستقیم و خالص هم‌پوشانی» نامید.

طرح. دنباله‌ی درخواست، وظیفه‌ی هدف و موقعیت آن دقیقا ثابت نگه داشته می‌شوند و تنها *پراکندگی* زیرشبکه تغییر می‌کند. پراکندگی تعیین می‌کند هر ماسک وظیفه چند مختصه بگیرد و در نتیجه با $|S_{\text{share}}| =$

$|M_f \cap M_r|$ ارتباط مستقیم دارد، اما هم‌زمان ظرفیت هر وظیفه را نیز تغییر می‌دهد. پنج سطح پراکندگی از ۰/۹ تا ۰/۵، دو روش و پنج بذر، در مجموع ۱۰۰ اجرای هم‌تراز می‌سازند. این جاروب افراز و دنباله‌ی پروتکل استاندارد را به‌کار می‌برد، اما با بودجه‌ی ۳ ایپاک به‌ازای هر وظیفه اجرا شده است و نه ۲۰ ایپاک. بنابراین سطح مطلق دقت آن با جدول‌های بخش ۴-۳ قابل مقایسه نیست و تنها مقایسه‌ی *درون‌جدولی* دو بازو در هر سطح پراکندگی معتبر است. دو بازوی هر خانه پراکندگی و بذر یکسان می‌بینند؛ هم‌پوشانی محقق‌شده‌ی آن‌ها کاملا برابر نیست، ازاین‌رو میانگین جفت گزارش می‌شود. بیشترین اختلاف نسبی دو بازو کمتر از ٪۰/۰۵ است.

محک جاروب. برخلاف محک موقعیت درخواست که متغیر مستقلش تنها سه یا چهار مقدار یگانه داشت، اینجا هم‌پوشانی اندازه‌گیری‌شده در بازه‌ای ۱۹/۷ برابری روی CIFAR-10 و ۱۰/۷ برابری روی CIFAR-100 حرکت می‌کند و ترتیب آن با سطوح پراکندگی یکنواست. این مشاهده نشان می‌دهد دستگیره‌ی پراکندگی، دامنه‌ی کافی در هم‌پوشانی ایجاد کرده است؛ بااین‌حال اثر هم‌پوشانی را از اثر ظرفیت جدا نمی‌کند.

جدول ۴-۷: جاروب پراکندگی زیرشبکه و هم‌پوشانی ساختاری اندازه‌گیری‌شده (پنج بذر در هر خانه). متغیر دستکاری‌شده *پراکندگی* است، نه خود هم‌پوشانی؛ بنابراین این جدول یک دستکاری علّی خالصِ هم‌پوشانی نیست. دنباله‌ی درخواست، وظیفه‌ی هدف و موقعیت آن ثابت‌اند، اما پراکندگی هم $|S_{\text{share}}|$ و هم ظرفیت هر وظیفه را تغییر می‌دهد. مقدار $|S_{\text{share}}|$ میانگین دو بازوی هم‌بذر است؛ اختلاف نسبی دو بازو در همه‌ی جفت‌ها کمتر از ٪۰/۰۵ بود. «سود حفاظت» اختلاف بیشترین افت PALL منهای EPALL است؛ مقدار مثبت یعنی حفاظت کمک می‌کند. ستون دقت PALL محک تغییر ظرفیت است.

مجموعه‌داده	پراکندگی	$\lvert S_{\text{share}}\rvert$	بیشترین افت PALL	بیشترین افت EPALL	سود حفاظت	دقت PALL
	0.9	169,815	0.0036	0.0018	+0.0018	0.8855
	0.8	642,863	0.0121	0.0030	+0.0091	0.8826
CIFAR-10	0.7	1,365,591	0.0141	0.0012	+0.0129	0.8749
	0.6	2,285,622	0.0326	0.0040	+0.0286	0.8546
	0.5	3,347,700	0.0411	0.0062	+0.0349	0.8379
	0.9	959,093	0.0176	0.0030	+0.0146	0.4919
	0.8	3,014,623	0.0610	0.0126	+0.0484	0.4759
CIFAR-100	0.7	5,429,857	0.1124	0.0360	+0.0764	0.4160
	0.6	7,882,273	0.1454	0.0452	+0.1002	0.3577
	0.5	10,264,556	0.1550	0.0756	+0.0794	0.3292

جدول ۴-۷ نتیجه را نشان می‌دهد. روی CIFAR-10 سود حفاظت از ۰/۰۰۱۸ در کم‌ترین هم‌پوشانی به ۰/۰۳۴۹ در بیش‌ترین می‌رسد. روی CIFAR-100 از ۰/۰۱۴۶ آغاز می‌شود، در سطح پراکندگی ۰/۶ به بیشینه‌ی ۰/۱۰۰۲ می‌رسد و در بیشترین هم‌پوشانی ۰/۰۷۹۴ است؛ بنابراین روند مثبت است اما یکنوا نیست. همبستگی رتبه‌ای اسپیرمن میان هم‌پوشانی اندازه‌گیری‌شده و سود حفاظت روی CIFAR-10 برابر ۱/۰۰+ و روی CIFAR-100 برابر ۰/۹۰+ است. چون تنها پنج سطح در کار است، به‌جای تقریب مجانبی، هر ۱۲۰ جایگشت ممکن شمرده شده و p دقیق به‌دست آمده است: به‌ترتیب ۰/۰۰۸۳ و ۰/۰۴۱۷. برای سود دقت نهایی، همبستگی روی هر دو مجموعه‌داده ۱/۰۰+ و p دقیق ۰/۰۰۸۳ است.

دو قید. نخست، پراکندگی علاوه بر هم‌پوشانی، *ظرفیت* را هم تغییر می‌دهد و این مخدوش‌کننده با طراحی حذف‌شدنی نیست: دقت PALL در همین جاروب روی CIFAR-100 از ۰/۴۹۱۹ به ۰/۳۲۹۲ می‌افتد. همسان‌بودن پراکندگی اسمی در هر جفت، مقایسه‌ی دو روش را تفسیرپذیرتر می‌کند، اما اثر علّی مستقل هم‌پوشانی را شناسایی نمی‌کند. دوم، آزمون علامت بذری درون هر خانه با پنج بذر توان کمی دارد و تنها سه خانه از ده خانه به $p < ۰/۰۵$ می‌رسند؛ میانگین‌ها اختلاف بزرگی نشان می‌دهند اما پراکندگی بذرها بالاست. بنابراین شاهد اصلی این بخش در روند میان سطوح است و نه در تک‌تک خانه‌ها.

جمع‌بندی. با ثابت‌نگه‌داشتن هویت و موقعیت وظیفه‌ی هدف، جاروب پراکندگی یک رابطه‌ی رتبه‌ای مثبت میان هم‌پوشانی اندازه‌گیری‌شده و سود حفاظت نشان می‌دهد. این نتیجه از ادعای انگیزشی EPALL پشتیبانی می‌کند، اما به‌دلیل تغییر هم‌زمان ظرفیت و غیریکنوایی CIFAR-100 اثبات علّی خالصِ اثر هم‌پوشانی نیست.

۴-۶ انتساب سازوکار EPALL

پرسش RQ4 می‌پرسد کدام جزء سازوکار واقعا منشأ بهبود است. جدول ۴-۸ پنج بازوی هم‌تراز را نشان می‌دهد که پروتکل هم‌پوشانی‌سنگین، دنباله‌ها و بذرها را به اشتراک می‌گذارند و هر یک تنها در یک جزء تفاوت دارند؛ تعریف این بازوها در بخش ۳-۳-۵ آمد.

جدول ۴-۸: کنترل‌های مستقیم سازوکار EPALL روی هشت بذر، زیر پروتکل هم‌پوشانی‌سنگین و با دنباله‌های هم‌تراز. هر بازو دقیقا یک جزء سازوکار محافظت را تغییر می‌دهد و هیچ سلولی رتبه‌بندی نشده است. سطح تصادفی دقت هدف ۰/۵ برای C10 و ۰/۲ برای C100 است.

داده	بازوی کنترل	دقت نهایی	فراموشی	بیشترین افت	$\lvert A_u - c\rvert$
C10	بدون جریمه‌ی ℓ_2	0.8822 ± 0.0358	0.2533 ± 0.0233	0.0025 ± 0.0029	0.0162 ± 0.0272
	فقط هم‌پوشانی	0.8767 ± 0.0429	0.2503 ± 0.0232	-0.0001 ± 0.0002	0.0127 ± 0.0203
	بودجه‌ی تصادفی	0.8747 ± 0.0383	0.2547 ± 0.0218	0.0097 ± 0.0107	0.0146 ± 0.0233
	رتبه‌بندی بدون اشتراک	0.8774 ± 0.0394	0.2502 ± 0.0222	0.0021 ± 0.0027	0.0136 ± 0.0281
	کامل	0.8774 ± 0.0394	0.2502 ± 0.0222	0.0021 ± 0.0027	0.0136 ± 0.0281
C100	بدون جریمه‌ی ℓ_2	0.4877 ± 0.0186	0.0889 ± 0.0179	0.0082 ± 0.0082	0.0083 ± 0.0084
	فقط هم‌پوشانی	0.4808 ± 0.0205	0.0904 ± 0.0186	0.0005 ± 0.0009	0.0080 ± 0.0077
	بودجه‌ی تصادفی	0.4897 ± 0.0197	0.0944 ± 0.0214	0.0227 ± 0.0082	0.0098 ± 0.0098
	رتبه‌بندی بدون اشتراک	0.4929 ± 0.0236	0.0923 ± 0.0206	0.0080 ± 0.0081	0.0155 ± 0.0170
	کامل	0.4929 ± 0.0236	0.0923 ± 0.0206	0.0080 ± 0.0081	0.0155 ± 0.0170

جدول ۴-۹ اختلاف‌های جفتی نسبت به بازوی کامل را می‌دهد و خوانش دقیق‌تری ممکن می‌سازد.

جدول ۴-۹: اختلاف‌های جفتی هر بازوی کنترل نسبت به بازوی کامل؛ مقدار مثبت به سود بازوی کامل است. بازه‌های بوت‌استرپ روی هشت بذر هم‌تراز و توصیفی‌اند.

داده	بازوی کنترل	Δ دقت نهایی	Δ بیشترین افت	$\Delta\lvert A_u - c\rvert$
CIFAR-10	بدون جریمه‌ی ℓ_2	-0.0048 $[-0.0113, 0.0008]$	0.0004 $[-0.0011, 0.0022]$	0.0026 $[-0.0007, 0.0083]$
	فقط هم‌پوشانی	0.0007 $[-0.0027, 0.0042]$	-0.0022 $[-0.0041, -0.0006]$	-0.0009 $[-0.0094, 0.0054]$
	بودجه‌ی تصادفی	0.0027 $[0.0009, 0.0042]$	0.0076 $[0.0024, 0.0134]$	0.0010 $[-0.0056, 0.0093]$
	رتبه‌بندی بدون اشتراک	0.0000 $[0.0000, 0.0000]$	0.0000 $[0.0000, 0.0000]$	0.0000 $[0.0000, 0.0000]$
CIFAR-100	بدون جریمه‌ی ℓ_2	0.0052 $[-0.0008, 0.0129]$	0.0002 $[-0.0078, 0.0083]$	-0.0072 $[-0.0170, 0.0038]$
	فقط هم‌پوشانی	0.0120 $[0.0065, 0.0191]$	-0.0075 $[-0.0127, -0.0022]$	-0.0075 $[-0.0160, -0.0000]$
	بودجه‌ی تصادفی	0.0031 $[-0.0029, 0.0091]$	0.0148 $[0.0102, 0.0210]$	-0.0057 $[-0.0137, 0.0018]$
	رتبه‌بندی بدون اشتراک	0.0000 $[0.0000, 0.0000]$	0.0000 $[0.0000, 0.0000]$	0.0000 $[0.0000, 0.0000]$

خوانش این دو جدول چهار نتیجه می‌دهد. دو نتیجه به سود سازوکار پیشنهادی است و دو نتیجه نیست؛ هر چهار به‌صراحت گزارش می‌شوند.

یک. کیفیت انتخاب مهم است، نه فقط اندازه‌ی بودجه. این روشن‌ترین و پایدارترین یافته‌ی کنترل‌ها است و روی هر *دو* مجموعه‌داده برقرار می‌ماند. بازوی بودجه‌ی تصادفی همان تعداد مختصه را محافظت می‌کند، اما بیشترین افت را ۰/۰۰۷۶ روی CIFAR-10 و ۰/۰۱۴۸ روی CIFAR-100 بدتر می‌کند و هر دو بازه صفر را در بر نمی‌گیرند؛ روی CIFAR-10 اختلاف دقت نیز (+۰/۰۰۲۷) از صفر جدا است. این مقایسه دقیقا همان چیزی را جدا می‌کند که باید: با ثابت‌نگه‌داشتن اندازه‌ی بودجه، تفاوت باقی‌مانده تنها می‌تواند از کیفیت رتبه‌بندی بیاید.

دو. رتبه‌بندی نسبت به محافظت بدون رتبه، نگه‌داری بهتری می‌دهد اما همه‌ی معیارها را غالب نمی‌کند. روی CIFAR-100 بازوی کامل دقت جفتی را نسبت به فقط‌هم‌پوشانی ۰/۰۱۲۰ [۰/۰۰۶۵, ۰/۰۱۹۱] بهبود می‌دهد. اما بازوی فقط‌هم‌پوشانی بیشترین افت *کمتری* دارد، و این بار روی هر دو مجموعه‌داده: ۰/۰۰۲۲ روی CIFAR-10 و ۰/۰۰۷۵ روی CIFAR-100، هر دو با بازه‌ای که صفر را در بر نمی‌گیرد. یعنی رتبه‌بندی، پایداری بی‌درنگ بدترین وظیفه را در ازای نگه‌داری نهایی بالاتر معاوضه می‌کند. این یک مصالحه‌ی واقعی است و نه یک برتری یکنواخت؛ اگر معیار عملیاتی یک سامانه بدترین وظیفه باشد، محافظت بدون رتبه‌بندی انتخاب بهتری است.

سه. سهم مستقل جریمه‌ی ℓ_2 پشتیبانی نمی‌شود. این بازو تنها جمله‌ی جریمه را حذف می‌کند؛ شناسایی ناحیه‌ی بحرانی و ساخت پشتیبان‌های آن دست‌نخورده باقی می‌ماند، پس نتیجه‌ی آن درباره‌ی *شدت مهار* است و نه درباره‌ی کل محافظت. این بازو روی هیچ‌یک از دو مجموعه‌داده به‌روشنی از بازوی کامل جدا نمی‌شود: اختلاف دقت جفتی روی CIFAR-100 برابر ۰/۰۰۵۲ با بازه‌ی [−۰/۰۰۰۸, ۰/۰۱۲۹] است که صفر را در بر می‌گیرد، و روی CIFAR-10 برآورد نقطه‌ای ۰/۰۰۴۸− است، یعنی حتی کمی به سود *حذف* لنگر. اختلاف بیشترین افت در هر دو مورد عملا صفر است. بنابراین در پیکربندی آزمون‌شده، آنچه اثر دارد *انتخاب* ناحیه‌ی محافظت‌شده است و نه شدت مهار آن با جمله‌ی ℓ_2. این یافته با نسخه‌ی پنج‌بذری همین کنترل‌ها تفاوت دارد، جایی که بازه‌ی CIFAR-100 صفر را در بر نمی‌گرفت؛ افزودن بذر آن جدایی را از میان برد.

چهار. فیلتر صریح اشتراک در *این پیاده‌سازی* اضافی است. بازوی رتبه‌بندی‌بدون‌اشتراک، بازوی کامل را *دقیقا* بازتولید می‌کند؛ همه‌ی اختلاف‌های جفتی صفرند. اما تفسیر این یکسانی باید دقیق باشد و **ضرورت عمومی هم‌پوشانی را رد نمی‌کند.** دلیلش ساختاری است: در این بازو رتبه‌بندی روی کل زیرشبکه‌ی هدف M_f انجام می‌شود، ولی ماسک ترمیم *مستقل از حالت کنترل* همان $M_f \cap M_r$ می‌ماند و گرادیان بیرون آن صفر می‌شود. پس هر مختصه‌ای که رتبه‌بندی بیرون M_r انتخاب کند، در عمل هرگز حرکت نمی‌کند و لنگر

روی آن بی‌اثر است. به بیان دیگر، خود پیاده‌سازی فیلتر اشتراک را در پایین‌دست اعمال می‌کند؛ بنابراین این آزمایش نشان می‌دهد اعمال *دوباره‌ی* آن در بالادست اضافی است، نه اینکه محدودکردن به ناحیه‌ی مشترک بی‌فایده باشد.

جمع‌بندی کنترل‌ها. آنچه این کنترل‌ها به‌طور تجربی پشتیبانی می‌کنند، باریک‌تر از توصیف کامل سازوکار است: *رتبه‌بندی مختصه‌های کاندید بر پایه‌ی حساسیت نگه‌داری، با یک بودجه‌ی ثابت.* دو جزء دیگر سهم مستقل قابل‌اندازه‌گیری نشان نمی‌دهند، اما این به‌معنای بی‌فایده‌بودن آن‌ها نیست: جریمه‌ی ℓ_2 تنها شدت مهار را تنظیم می‌کند در حالی که پشتیبان‌ها مستقل از آن ساخته می‌شوند، و فیلتر اشتراک در پایین‌دست پیاده‌سازی به‌هرحال اعمال می‌شود. این کنترل‌ها برای اثبات *ضرورت* همه‌ی اجزای طراحی کافی نیستند. این محدودیت در بخش ۵-۴ به‌عنوان محدودیت ادعای سازوکاری ثبت می‌شود و در بخش ۵-۵ به یک پیشنهاد ساده‌سازی تبدیل می‌شود.

۴-۷ رژیم پیش‌آموزش‌دیده‌ی PEFT

جدول ۴-۱۰ فرض روش‌های کارآمد پارامتری را برقرار می‌کند: LoRA و PALL-Adapter هر دو روی یک ResNet-18 منجمد و پیش‌آموزش‌دیده‌ی ImageNet می‌نشینند. ورودی‌های CIFAR از نرمال‌سازی مجموعه‌داده به آماره‌های ImageNet که بدنه انتظار دارد تبدیل می‌شوند؛ TinyImageNet از پیش همان آماره‌ها را دارد. این تصحیح نرمال‌سازی مهم است و در پیوست ۵-۶ توضیح داده شده است.

جدول ۴-۱۰: رژیم تصحیح‌شده‌ی بدنه‌ی منجمد پیش‌آموزش‌دیده (سه بذر در همه‌ی سطرها). ورودی CIFAR با آماره‌های ImageNet نرمال می‌شود و TinyImageNet از پیش همان آماره‌ها را دارد. سطح تصادفی دقت هدف به‌ترتیب ۰/۵، ۰/۲ و ۰/۱ است. پرنگ تنها نقطه‌ی پایان بهتر در دقت نهایی و بیشترین افتِ علامت‌دار را نشان می‌دهد؛ دقت هدف با فاصله تا سطح تصادفی خوانده می‌شود. R_{upd} نسبت اسمی پارامترهای آموزش‌پذیر در ماسک ترمیم است.

مجموعه‌داده	روش	دقت نهایی	بیشترین افت	دقت هدف	R_{upd}
CIFAR-10	LoRA	**0.9848 ± 0.0154**	0.0000 ± 0.0000	0.5203 ± 0.0966	0.0011
	Adapter	0.9841 ± 0.0172	**−0.0002 ± 0.0003**	0.5190 ± 0.0988	0.0030
CIFAR-100	LoRA	0.8215 ± 0.0083	0.0213 ± 0.0219	0.1613 ± 0.0613	0.0051
	Adapter	**0.8277 ± 0.0061**	**0.0180 ± 0.0020**	0.1907 ± 0.1066	0.0103
TinyImageNet	LoRA	0.8980 ± 0.0049	0.0153 ± 0.0110	0.0940 ± 0.0092	0.0163
	Adapter	**0.9047 ± 0.0050**	**0.0127 ± 0.0042**	0.0940 ± 0.0401	0.0273

با بدنه‌ی پیش‌آموزش‌دیده، هر دو روش به‌طور چشمگیری بهبود می‌یابند: PALL-Adapter به ۰/۹۸۴۱ روی CIFAR-10، ۰/۸۲۷۷ روی CIFAR-100 و ۰/۹۰۴۷ روی TinyImageNet می‌رسد و مقادیر متناظر LoRA برابر ۰/۹۸۴۸، ۰/۸۲۱۵ و ۰/۸۹۸۰ است. اختلاف CIFAR-10 نسبت به پراکندگی بذرها ناچیز است، در حالی که آداپتر روی CIFAR-100 و TinyImageNet به‌طور متوسط کمی بالاتر می‌ایستد. مقایسه‌ی این نقطه‌های پایان با هم، مقایسه‌ی رفتار انتهایی دو معماری است و تفاوت را به ماسک نرم نسبت نمی‌دهد؛ آن پرسش، موضوع بخش بعد است.

این جدول در کنار جدول‌های ۴-۱ و ۴-۲ همان نکته‌ی مهم را روشن می‌کند: عملکرد پایین دو سطر PEFT در پروتکل از صفر، ناشی از سازوکار فراموشی آن‌ها نبود، بلکه ناشی از نشستن روی یک بازنمایی تصادفی بود.

۴-۸ انتساب مؤلفه‌های PALL-Adapter: یک نتیجه‌ی منفی

این بخش به RQ5 پاسخ می‌دهد و پاسخ، منفی است. پنج بازوی هم‌تراز زیر خط لوله‌ی پیش‌آموزش‌دیده‌ی تصحیح‌شده و با دنباله‌های یکسان (سه بذر) مؤلفه‌های زمان درخواست را جدا می‌کنند: *فقط بازنشانی* تنها آداپتر هدف و برش طبقه‌بند آن را بازنشانی می‌کند؛ *بازنشانی+ترمیم* ترمیم نگه‌داری را اضافه می‌کند؛ *یکنواخت* محافظت هم‌پوشانی را از به‌روزرسانی مشترک برمی‌دارد؛ *ماسک/بدون‌صعود* گام صعود ذخیره‌شده‌ی

طبقه‌بند را حذف می‌کند؛ و *کامل* همه‌ی مؤلفه‌ها را دارد. جدول ۴-۱۱ نتیجه‌ی این پنج بازو را می‌دهد.

جدول ۴-۱۱: کنترل‌های مؤلفه‌ای زمان درخواست برای PALL-Adapter زیر خط لوله‌ی تصحیح‌شده‌ی پیش‌آموزش‌دیده و دنباله‌های هم‌تراز (سه بذر). بازوها از نظر مولد اعداد تصادفی هم‌ترازند، پس یکسان‌بودن مقادیر نشانه‌ی مؤلفه‌ی واقعا بی‌اثر است و نه نوفه. سطح تصادفی دقت هدف ۰/۵ برای C10 و ۰/۲ برای C100 است.

داده	بازوی کنترل	دقت نهایی	بیشترین افت	$\lvert A_u - c \rvert$	T_f (ث)
C10	فقط بازنشانی	0.9836 ± 0.0168	0.0000 ± 0.0000	0.0675 ± 0.0573	0.0025
	بازنشانی + ترمیم	0.9840 ± 0.0171	0.0002 ± 0.0003	0.0677 ± 0.0569	0.8529
	یکنواخت (بی‌محافظت)	0.9840 ± 0.0171	0.0000 ± 0.0000	0.0667 ± 0.0577	0.9885
	ماسک، بدون صعود	0.9841 ± 0.0170	0.0000 ± 0.0000	0.0675 ± 0.0573	0.9991
	کامل	0.9841 ± 0.0170	0.0000 ± 0.0000	0.0658 ± 0.0586	0.9849
C100	فقط بازنشانی	0.8324 ± 0.0076	0.0000 ± 0.0000	0.0787 ± 0.0163	0.0021
	بازنشانی + ترمیم	0.8248 ± 0.0073	0.0187 ± 0.0070	0.0880 ± 0.0251	2.8273
	یکنواخت (بی‌محافظت)	0.8250 ± 0.0079	0.0153 ± 0.0196	0.0827 ± 0.0260	3.3232
	ماسک، بدون صعود	0.8244 ± 0.0091	0.0147 ± 0.0202	0.0887 ± 0.0212	3.1549
	کامل	0.8244 ± 0.0091	0.0147 ± 0.0202	0.0847 ± 0.0200	3.3178

سه یافته از این جدول بیرون می‌آید.

ماسک نرم در نقطه‌ی پایان بی‌اثر است. بازوی کامل و بازوی یکنواخت حداکثر ۰/۰۰۰۶ در A_{final} یا بیشترین افت با هم اختلاف دارند، و بازوی کامل و ماسک/بدون‌صعود در هر دو معیار *یکسان*اند. چون بازوها از نظر مولد اعداد تصادفی هم‌تراز طراحی شده‌اند، این یکسانی نشانه‌ی نوفه نیست؛ نشانه‌ی یک مؤلفه‌ی واقعا بی‌اثر است.

صعود طبقه‌بند تنها مؤلفه‌ی با اثر قابل‌اندازه‌گیری است. حذف آن، نگه‌داری را دست‌نخورده می‌گذارد اما وظیفه‌ی فراموش‌شده را به‌اندازه‌ی ۰/۰۰۴۰ روی CIFAR-100 و ۰/۰۰۱۷ روی CIFAR-10 از سطح تصادفی دور می‌کند. یعنی این مؤلفه در سرکوب سهم دارد، هرچند کوچک، و به دقت نگه‌داری آسیب نمی‌زند.

روی CIFAR-100 فقط‌بازنشانی در واقع بهتر است. نگه‌داری ۰/۸۳۲۴ در برابر ۰/۸۲۴۴، بیشترین افت ۰ در برابر ۰/۰۱۴۷، و فاصله‌ی میانگین کوچک‌تری تا سطح تصادفی. بازه‌های بوت‌استرپ جفتی روی سه دنباله برای اختلاف دقت و اختلاف فاصله تا سطح تصادفی صفر را در بر می‌گیرند، در حالی که اختلاف بیشترین افت به سود فقط‌بازنشانی است.

نتیجه‌گیری صریح: سرکوب مشاهده‌شده در مسیر PALL-Adapter عمدتا به *بازنشانی مسیر هدف* نسبت داده می‌شود. به‌روزرسانی مشترکِ آگاه از هم‌پوشانی در این مسیر، مؤلفه‌ای است که از نظر نظری دامنه‌ی مشخصی دارد (بخش ۳-۵) اما در نقطه‌ی پایان اندازه‌گیری‌شده پشتوانه‌ی تجربی ندارد. به همین دلیل، این پایان‌نامه مسیر آداپتر را یک گزینه‌ی اکتشافی ممیزی‌شده می‌داند و آن را به‌عنوان مشارکت دوم ارائه نمی‌کند.

۴-۸-۱ ماسک نرم پیوسته: جاروب شدت

نتیجه‌ی منفی بخش پیش، ماسک نرم گسسته را ارزیابی می‌کند: پارتیشن دودویی کامل/نرم/منجمد با یک ضریب ثابت. یک تبیین محتمل برای آن نتیجه این است که مشکل نه در ایده‌ی محافظت، بلکه در درشت‌*دانگی* پارتیشن باشد. برای آزمودن این تبیین، نسخه‌ی پیوسته‌ای پیاده‌سازی و جاروب شد که در آن هر مختصه‌ی زیرمجموعه‌ی فراموشی ضریب خاص خود را می‌گیرد:

$$m_i = \mathrm{clamp}\big(۱ - \gamma\, \hat{c}_i,\ ۰,\ ۱\big), \qquad \hat{c}_i = \frac{\mathrm{relu}\big(-\, g_i^f\, g_i^r\big)}{\max_j \mathrm{relu}\big(-\, g_j^f\, g_j^r\big)}, \tag{۳-۴}$$

که در آن g^f و g^r گرادیان فراموشی و نگه‌داری روی آداپتر مشترک‌اند و $\hat{c}_i$ انرژی تعارض بهنجارشده است. مقدار $\gamma = ۰$ به‌روزرسانی کامل را بازمی‌گرداند، پس بازوی کنترل درون خود جاروب قرار دارد و این آزمایش به هیچ جدول دیگری تکیه نمی‌کند.

جدول ۴-۱۲: جاروب شدت γ در ماسک نرم پیوسته‌ی وزن‌دار به انرژی تعارض، روی پروتکل پیش‌آموزش‌دیده (سه بذر). ضریب هر مختصه $m_i = \mathrm{clamp}(۱ - \gamma \hat{c}_i, ۰, ۱)$ است و $\gamma = ۰$ به‌روزرسانی کامل را بازمی‌گرداند، پس بازوی کنترل درون همین جدول است. ستون‌های سمت راست وضعیت پایانی و ستون‌های سمت چپ آمار خودِ ماسک در لحظه‌ی درخواست‌اند؛ این دو باید کنار هم خوانده شوند، چون معنای یک نتیجه‌ی پوچ بسته به آنکه ماسک تغییر کرده باشد یا نه، کاملا متفاوت است. «٪تعارض» سهم مختصه‌های زیرمجموعه‌ی فراموشی با انرژی تعارض ناصفر و «$m < ۰/۵$٪» سهم مختصه‌هایی است که ضریبشان واقعا زیر یک‌دوم افتاده است.

مجموعه‌داده	γ	دقت نهایی	بیشترین افت	دقت هدف	کامل	نرم	$m < ۰/۵$٪	٪تعارض
	0	0.9840	0.0000	0.5210	4916	0	0.00	40.62
	0.25	0.9840	0.0000	0.5210	3103	1813	0.00	40.74
CIFAR-10	0.5	0.9840	0.0000	0.5208	3069	1847	0.00	40.74
	1	0.9840	0.0000	0.5208	3044	1872	0.07	40.68
	2	0.9840	0.0000	0.5210	3026	1886	0.28	40.66
	0	0.8250	0.0153	0.1893	4916	0	0.00	44.13
	0.25	0.8250	0.0147	0.1873	2729	2187	0.00	44.50
CIFAR-100	0.5	0.8251	0.0153	0.1907	2714	2202	0.00	44.80
	1	0.8252	0.0153	0.1887	2747	2168	0.19	44.12
	2	0.8256	0.0147	0.1900	2723	2183	0.85	44.62

جدول ۴-۱۲ هم وضعیت پایانی و هم آمار خودِ ماسک را کنار هم می‌دهد؛ این دو باید با هم خوانده شوند، چون معنای یک نتیجه‌ی پوچ بسته به آنکه ماسک تغییر کرده باشد یا نه کاملا متفاوت است.

وضعیت پایانی حرکت نمی‌کند. در سراسر جاروب، دامنه‌ی تغییر دقت نهایی روی CIFAR-10 صفر و روی CIFAR-100 تنها ۰/۰۰۰۶ است؛ بیشترین افت و دقت هدف نیز در همین حد جابه‌جا می‌شوند. با سه بذر، هیچ‌کدام از این دامنه‌ها از نوفه‌ی بذر قابل‌تفکیک نیستند.

اما ماسک واقعا تغییر می‌کند. این نکته، تفاوت میان یک نتیجه‌ی منفی معنادار و یک اشکال پیاده‌سازی است. در $\gamma = ۰$ هر ۴۹۱۶ مختصه‌ی زیرمجموعه‌ی فراموشی به‌روزرسانی کامل می‌گیرند، و از $\gamma = ۰/۲۵$ به بعد حدود یک‌سوم آن‌ها به دسته‌ی «نرم» منتقل می‌شوند. سیگنال تعارض هم وجود دارد و ساختگی نیست: میان ۴۰ تا ۴۵ درصد مختصه‌ها انرژی تعارض ناصفر دارند. پس نه ماسک ثابت مانده و نه سیگنال غایب است.

تبیین: دنباله‌ی بسیار نازک. توضیح در توزیع $\hat{c}$ نهفته است. این کمیت با بیشینه بهنجار می‌شود، پس $\max_i \hat{c}_i = ۱$ است؛ اما *میانگین* آن تنها ۰/۰۰۱۰۶ روی CIFAR-10 و حدود ۰/۰۰۴ روی CIFAR-100 است، یعنی سه مرتبه‌ی بزرگی کمتر از بیشینه. انرژی تعارض بنابراین در دنباله‌ای بسیار نازک متمرکز است و اکثریت قاطع مختصه‌ها $\hat{c}_i \approx ۰$ دارند. با ضریب *خطی* رابطه‌ی (۴-۳)، چنین مختصه‌ای حتی در بزرگ‌ترین γ آزموده‌شده ضریبی نزدیک به یک می‌گیرد. سهم مختصه‌هایی که ضریبشان واقعا زیر یک‌دوم می‌افتد، در $\gamma = ۲$ هنوز کمتر از یک درصد است. برچسب «نرم» در جدول بنابراین گمراه‌کننده است: آن مختصه‌ها اسما نرم‌اند اما عددا تقریبا کامل به‌روز می‌شوند.

نتیجه. تبیین درشت‌دانگی رد می‌شود. ریزکردن پارتیشن به یک ضریب پیوسته‌ی هر‌ـ‌مختصه، رفتار انتهابه‌انتها را تغییر نمی‌دهد، و دلیلش نیز مشخص است: با این صورت‌بندی، γ باید دو تا سه مرتبه‌ی بزرگی بزرگ‌تر باشد تا مختصه‌ی نوعی را به‌طور محسوس تضعیف کند. این یافته نتیجه‌ی منفی بخش پیش را تقویت می‌کند و هم‌زمان یک مسیر مشخص برای کارهای آینده باز می‌گذارد: ماسکی که به‌جای *مقدار* $\hat{c}$ بر *رتبه‌ی* آن بنا شود، از این مشکل مقیاس مصون است. ضمنا این جاروب همان سنجه‌ی جهت‌داری را که در بخش ۴-۱۱-۳ غایب بود فراهم می‌کند: نسبت هنجار گرادیان تنها بزرگی را می‌سنجید، در حالی که $\hat{c}$ از ضرب داخلی مختصه‌ای ساخته شده و جهت را در بر دارد.

۴-۹ نسخه‌های PALL زیر پروتکل هم‌پوشانی‌سنگین

این پروتکل از صفر و با سه ایپاک به‌ازای هر وظیفه، تنها سه نسخه‌ی PALL را مقایسه می‌کند و رژیم یا دنباله‌ی پروتکل استاندارد را به اشتراک نمی‌گذارد؛ بنابراین اعداد آن با جدول‌های ۴-۱ و ۴-۲ مخلوط نمی‌شوند. جدول ۴-۱۳ این مقایسه را نشان می‌دهد.

جدول ۴-۱۳: نسخه‌های PALL در رژیم مجزای هم‌پوشانی‌سنگین و از صفر (۳ ایپاک به‌ازای هر وظیفه). سطرهای CIFAR سه بذر دارند؛ سطرهای TinyImageNet از برچسب مجزای خود می‌آیند و توصیفی‌اند. سطح تصادفی دقت هدف به‌ترتیب ۰/۵، ۰/۲ و ۰/۱ است. اعداد این جدول با جدول‌های ۴-۱ و ۴-۲ مخلوط نمی‌شوند.

مجموعه‌داده	روش	بذر	دقت نهایی	بیشترین افت	دقت هدف
CIFAR-10	PALL	3	0.9087 ± 0.0266	0.0040 ± 0.0069	0.5000 ± 0.0000
	EPALL	3	0.9152 ± 0.0291	0.0008 ± 0.0033	0.5000 ± 0.0000
	PALL-Adapter	3	0.7829 ± 0.0610	0.0107 ± 0.0101	0.4802 ± 0.0816
CIFAR-100	PALL	3	0.5013 ± 0.0107	0.0187 ± 0.0095	0.2033 ± 0.0058
	EPALL	3	0.5093 ± 0.0148	0.0047 ± 0.0050	0.2033 ± 0.0058
	PALL-Adapter	3	0.3284 ± 0.0257	0.0427 ± 0.0333	0.2147 ± 0.0261
TinyImageNet	PALL	3	0.3365	0.0853	0.0973
	EPALL	3	0.4533	0.0840	0.0893
	PALL-Adapter	3	0.3332	0.0060	0.0993

روی هر دو مجموعه‌ی CIFAR، EPALL بالاترین دقت نگه‌داری میان سه نسخه را دارد (۰/۹۱۵۲ در برابر ۰/۹۰۸۷ و ۰/۵۰۹۳ در برابر ۰/۵۰۱۳) و هم‌زمان بیشترین افت را از ۰/۰۰۴۰ به ۰/۰۰۰۸ و از ۰/۰۱۸۷ به ۰/۰۰۴۷ کاهش می‌دهد. PALL-Adapter در جهت مخالف معاوضه می‌کند: دامنه‌ی ترمیم کوچک‌تر و A_u نزدیک به سطح تصادفی، به بهای دقت نگه‌داری وقتی آموزش از صفر آغاز می‌شود.

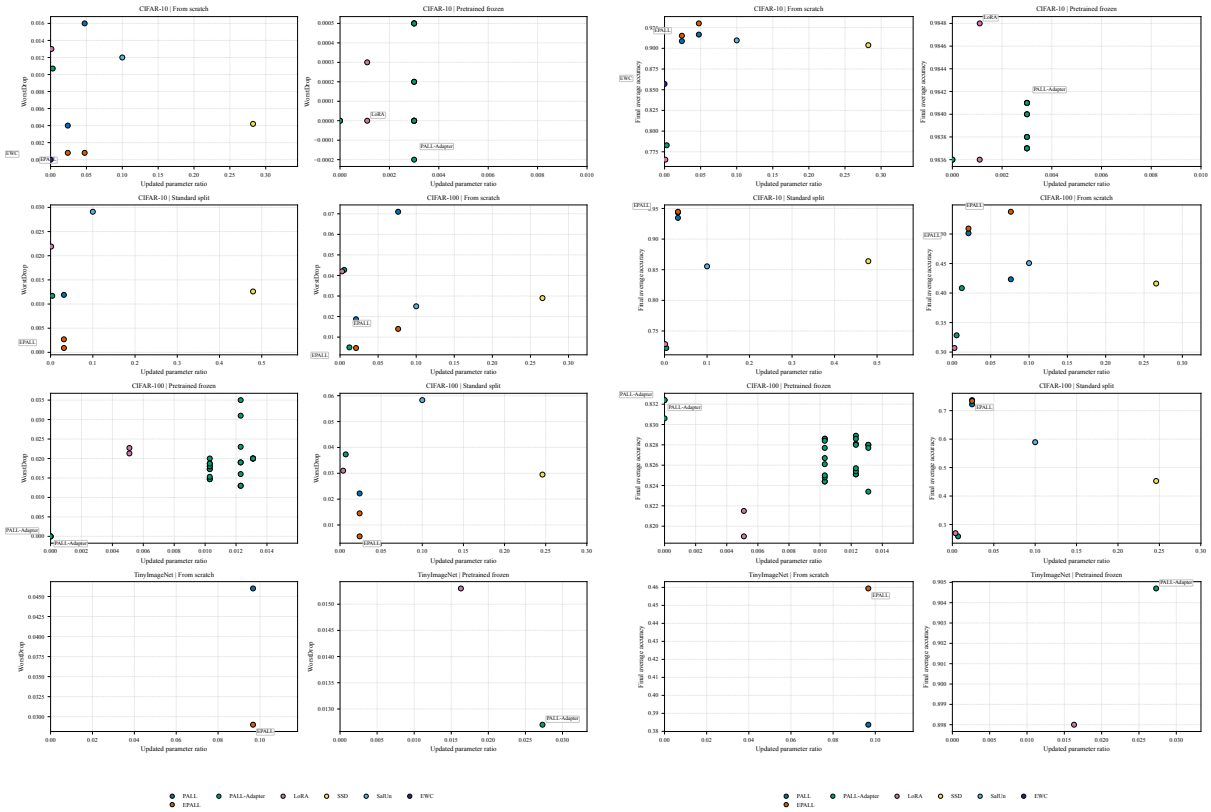
روی TinyImageNet — که سخت‌ترین محک این پایان‌نامه با ۲۰ وظیفه است — EPALL برتری دقت بزرگ‌تری نشان می‌دهد (۰/۴۵۳۳ در برابر ۰/۳۳۶۵). با این حال بیشترین افت هر دو نسخه‌ی کامل بالا می‌ماند (۰/۰۸۴۰ و ۰/۰۸۵۳)، که نشان می‌دهد در تنظیم بیست‌وظیفه‌ای و سه‌ایپاک، ظرفیت شبکه برای همه‌ی وظایف کافی نیست و مسئله دیگر فقط مسئله‌ی هم‌پوشانی نیست. بازه‌های میان‌بذری این سطرها هم‌پوشانی دارند، پس نتایج سه‌بذری این جدول روند توصیفی‌اند و رتبه‌بندی تاییدشده نیستند.

۴-۱۰ هزینه: دامنه‌ی ترمیم و حالت مقیم

۴-۱۰-۱ نسبت اسمی دامنه‌ی ترمیم

PALL-Adapter دامنه‌ی اسمی ترمیم نگه‌داری را نسبت به نسخه‌های تمام‌شبکه به‌شدت کاهش می‌دهد: از ۰/۰۲۴۰ به ۰/۰۰۳۰ روی CIFAR-10 (کاهش ۸۷/۵ درصدی)، از ۰/۰۲۱۰ به ۰/۰۰۵۵ روی CIFAR-100 (حدود ۷۴ درصد) و از ۰/۰۹۶۰ به ۰/۰۲۹۲ روی TinyImageNet (حدود ۷۰ درصد).

باید تصریح شود این نسبت‌ها چه چیزی را نمی‌سنجند. آن‌ها تنها دامنه‌ی پارامترهای آموزش‌پذیر در ماسک ترمیم را می‌پوشانند و بنابراین محاسبات انتهابه‌انتها، حافظه‌ی اوج، زمان دیواری یا تعداد کل مختصه‌های تغییریافته را اثبات نمی‌کنند؛ به‌علاوه، ذخیره‌سازی آداپترهای مخصوص وظیفه با تعداد وظایف رشد می‌کند. به همین دلیل این نسبت‌ها در این پایان‌نامه تشخیص پیاده‌سازی نامیده می‌شوند و نه معیار کارایی. شکل ۴-۵ این نسبت را در برابر دو معیار کیفیت نشان می‌دهد.



(آ) در برابر دقت نهایی

(ب) در برابر بیشترین افت

شکل ۴-۵: مصالحه‌ی میان نسبت اسمی دامنه‌ی ترمیم و دو معیار کیفیت، به تفکیک مجموعه‌داده و رژیم. محور افقی یک تشخیص پیاده‌سازی است و نه سنجه‌ی کارایی انتهابه‌انتها.

۴-۱۰-۲ حسابداری حالت مقیم

چون نسبت دامنه‌ی ترمیم یک اندازه‌گیری حافظه نیست، جدول ۴-۱۴ حالت تنسوری مقیم را روی دنباله‌های هم‌تراز بذر صفر به‌طور مستقیم حسابداری می‌کند. EPALL همه‌ی وظایف را در یک مدل مشترک با ماسک‌های پراکنده و پشتیبان‌ها نگه می‌دارد، در حالی که CLPU یک شبکه‌ی کامل به‌ازای هر وظیفه‌ی فعال دارد.

جدول ۴-۱۴: حسابداری حالت تنسوری مقیم روی دنباله‌های هم‌ترازِ بذر صفر، در بیشترین تعداد وظیفه‌ی فعال (بر حسب مبی‌بایت، 2^{20} بایت). این حسابداری پارامترهای مقیم، شبکه‌های جانبی CLPU، ماسک‌های زیرشبکه، اندیس و مقدار پشتیبان‌های پراکنده و تصویر و برچسب و لاجیت بازپخش را در بر می‌گیرد. حالت بهینه‌ساز، فعال‌سازی‌ها، سریال‌سازی و سربارهٔ مفسر را شامل نمی‌شود؛ پس یک مقایسه‌ی حالت پیاده‌سازی است و نه ادعایی درباره‌ی پاک‌شدن دیسک یا حافظه‌ی اوج.

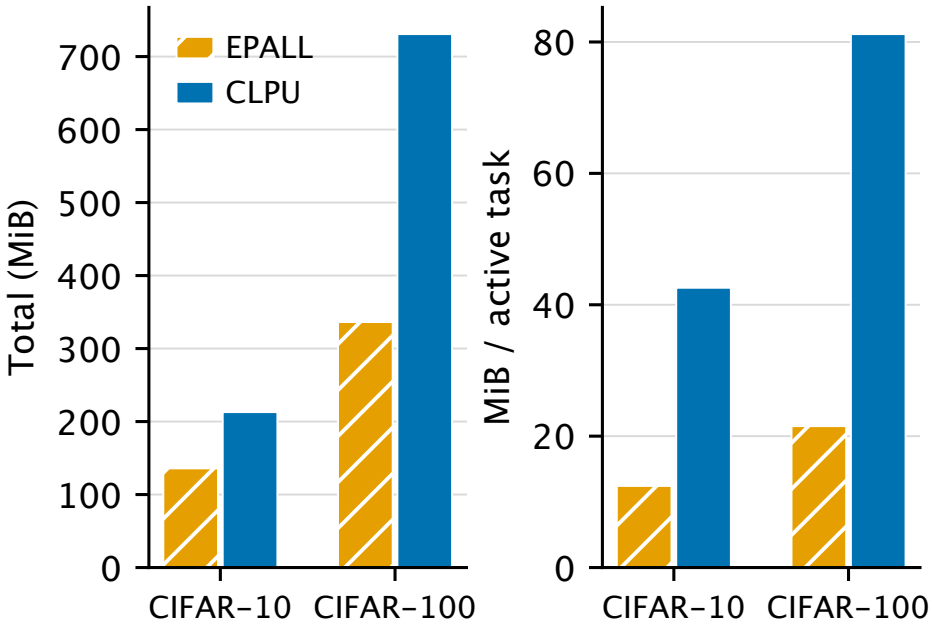
داده	روش	وظیفه‌ی فعال	مدل پایه	شبکه‌ی وظیفه	ماسک و پشتیبان	مجموع	به‌ازای وظیفه
CIFAR-10	**EPALL**	4	85.16	0.00	45.99	**135.85**	**12.42**
	CLPU	4	42.59	170.35	0.00	212.94	42.59
CIFAR-100	**EPALL**	8	162.30	0.00	169.29	**336.35**	**21.53**
	CLPU	8	81.20	649.58	0.00	730.78	81.20

CLPU در نتیجه میان ۱/۶ تا ۲/۲ برابر حالت حسابداری‌شده مصرف می‌کند و رشد آن به‌ازای هر وظیفه‌ی فعال ۳/۴ تا ۳/۸ برابر سریع‌تر است. این عدد، معنای واقعی «سقف بالا» بودن CLPU را روشن می‌کند: برتری کوچک آن در دقت، با هزینه‌ی حالت مقیمی خریده می‌شود که با تعداد وظایف خطی رشد می‌کند. شکل ۴-۶ همین ترکیب را نمایش می‌دهد.

۴-۱۱ ممیزی حریم خصوصی و ماندگاری

۴-۱۱-۱ حمله‌ی استنتاج عضویت اصلاح‌شده

جدول ۴-۱۵ نتیجه‌ی MIA افزایش‌داده‌ی هم‌تراز را نشان می‌دهد؛ در این نسخه نمونه‌های عضو و غیرعضو با پیش‌پردازش یکسان و بدون افزایش‌داده سنجیده می‌شوند و امتیازهای مساوی با رتبه‌ی میانگین تفکیک می‌شوند.



شکل ۴-۶: ترکیب حالت تنسوری مقیم در بیشترین تعداد وظیفه‌ی فعال. سهم EPALL میان مدل پایه، ماسک‌ها و پشتیبان‌های پراکنده تقسیم می‌شود، در حالی که سهم غالب CLPU از شبکه‌های جانبی هر-وظیفه می‌آید.

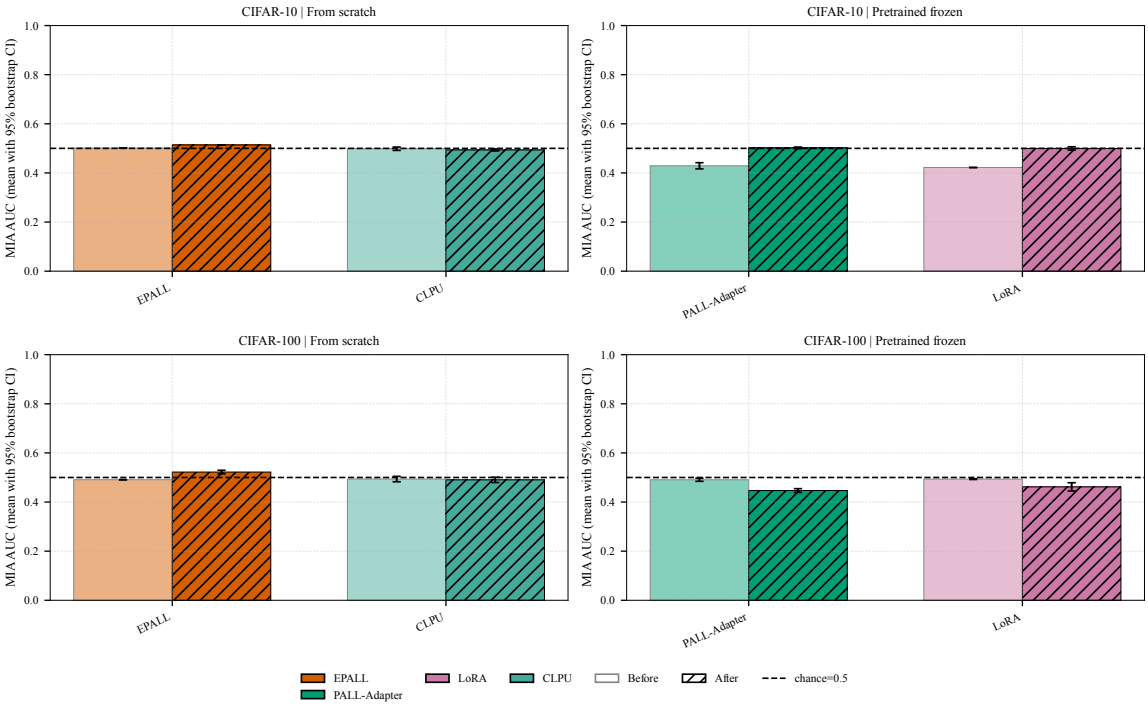
جدول ۴-۱۵: حمله‌ی استنتاج عضویت مبتنی بر آستانه‌ی زیان برای EPALL با پیش‌پردازش هم‌تراز (سه بذر). نمونه‌های عضو و غیرعضو پیش‌پردازش یکسان و بدون افزایش‌داده دارند و امتیازهای مساوی با رتبه‌ی میانگین تفکیک می‌شوند. مقدار نزدیک به ۰/۵ یک نتیجه‌ی تهیِ کم‌توان است و *شاهد حذف نیست*.

مجموعه‌داده	تعداد بذر	AUC پیش	AUC پس	Δ فاصله تا ۰/۵
CIFAR-10	3	0.5006 ± 0.0205	0.4959 ± 0.0155	0.0027 ± 0.0115
CIFAR-100	3	0.5015 ± 0.0169	0.4998 ± 0.0033	0.0098 ± 0.0088

هر دو مقدار پیش و پس از درخواست به ۰/۵ نزدیک‌اند. تفسیر درست این نتیجه محتاطانه است: حمله‌ی آستانه‌ی زیان در این تنظیم توان تفکیک کافی ندارد، پس نتیجه‌ی تهی به‌دست‌آمده *شاهد حذف* نیست و تنها می‌گوید این آزمون خاص چیزی پیدا نکرد. برای ادعای قوی‌تر، حمله‌های مبتنی بر مدل سایه [۱۱] یا ممیزی صوری [۲۰] لازم است. شکل ۴-۷ همین مقادیر را در سه رژیم نشان می‌دهد.

۴-۱۱-۲ آماره‌ی تفکیک‌پذیری حمله‌ی آستانه

نتیجه‌ی تهیِ بخش پیش تنها می‌گوید یک آزمون کم‌توان چیزی پیدا نکرد. سنجه‌ی حساس‌تر آن است که به‌جای AUC میانگین، بدترین نقطه‌ی کاری همان حمله را ببینیم. برای این کار عضو را کلاس مثبت



شکل ۴-۷: AUC حمله‌ی استنتاج عضویت پیش و پس از درخواست فراموشی، به تفکیک مجموعه‌داده و رژیم. خط افقی سطح ۰/۵ را نشان می‌دهد. نزدیکی به این خط، نتیجه‌ی تهیِ یک آزمون کم‌توان است.

می‌گیریم و روی شبکه‌ای از آستانه‌ها، کران پایین یک‌طرفه‌ی ۹۵ درصد کلاپر–پیرسون برای نرخ مثبت درست و کران بالای همان‌گونه برای نرخ مثبت نادرست را می‌سازیم؛ آنگاه

$$S = \max_{\text{آستانه}} \log \frac{\mathrm{TPR}_{\text{کران پایین}}}{\mathrm{FPR}_{\text{کران بالا}}}, \qquad (۴-۴)$$

که در صفر بریده می‌شود. **این کمیت را نباید پارامتر حریم خصوصی تفاضلی یا کران پایین آن خواند.** بازخورد بررسی، انجام یک حسابرسی تجربی حریم خصوصی و گزارش (ϵ, δ) را خواسته بود. آزمایش انجام شد و نتیجه‌ی آن در همین جدول است، اما خوانش (ϵ, δ) آن پذیرفته نشد؛ دلیلش این است که هرچند شکل ظاهری رابطه به ممیزی‌های صوری شبیه است، سه شرط لازم برقرار نیست: آستانه روی همان داده‌ای انتخاب می‌شود که آماره روی آن ارزیابی می‌شود، ساختار مجموعه‌داده‌های همسایه وجود ندارد، و شمول تصادفی و پارامتر δ نیز تعریف نشده‌اند. بنابراین S در این پایان‌نامه صرفا یک *آماره‌ی تفکیک‌پذیری* برای همین حمله‌ی خاص است. یک ممیزی صوری واقعی در سبک [۲۰] انجام نشده و در بخش ۵-۵ به‌عنوان کار آینده ثبت شده است. جدول ۴-۱۶ نتیجه را می‌دهد.

جدول ۴-۱۶: آماره‌ی تفکیک‌پذیری S پیش و پس از درخواست فراموشی، بر پایه‌ی امتیازهای هر-نمونه‌ی حمله‌ی عضویت و کران‌های کلاپر-پیرسون یک‌طرفه‌ی ۹۵ درصد. هر خانه روی شش رکورد (بذر $\times$ وظیفه‌ی حذف‌شده) خلاصه شده است. این کمیت پارامتر حریم خصوصی تفاضلی نیست و کران پایین معتبری برای آن هم نیست: آستانه روی همان داده انتخاب می‌شود و هیچ‌یک از شرط‌های مجموعه‌داده‌های همسایه، δ و شمول تصادفی برقرار نیست. مقدار بزرگ‌تر تنها یعنی همین حمله‌ی خاص، عضو و غیرعضو را بهتر تفکیک کرده است.

مجموعه‌داده	روش	میانگین S پیش	بیشینه‌ی S پیش	میانگین S پس	بیشینه‌ی S پس
CIFAR-10	**EPALL**	0.0000	0.0000	0.0993	0.4555
	PALL-Adapter	0.0005	0.0029	0.0058	0.0169
	LoRA	0.0002	0.0014	0.0080	0.0249
	CLPU	0.0023	0.0132	0.0000	0.0000
CIFAR-100	**EPALL**	0.0000	0.0000	0.0995	0.5544
	PALL-Adapter	0.0075	0.0263	0.0000	0.0000
	LoRA	0.0016	0.0076	0.0018	0.0108
	CLPU	0.0000	0.0000	0.0000	0.0000

این جدول یک یافته‌ی صریحا *نامساعد* برای روش پیشنهادی دارد و همان‌گونه گزارش می‌شود. پیش از درخواست، S برای EPALL روی هر دو مجموعه‌داده دقیقا صفر است؛ پس از درخواست، میانگین آن به ۰/۰۹۹۳ و ۰/۰۹۹۵ و بیشینه‌ی آن به ۰/۴۵۵۵ و ۰/۵۵۴۴ می‌رسد. یعنی همان عملیاتی که قرار بود اثر وظیفه‌ی هدف را بزداید، در این ممیزی نمونه‌های آن وظیفه را *تفکیک‌پذیرتر* کرده است. تفسیر محتاطانه‌ی این مشاهده چنین است: بازنشانی و ترمیم، رفتار مدل روی نمونه‌های وظیفه‌ی حذف‌شده را از الگوی نمونه‌های دیده‌نشده دور می‌کند و همین اختلاف، سیگنالی برای حمله می‌سازد. دقت هدف نزدیک به سطح تصادفی (جدول‌های ۴-۱ و ۴-۲) با این یافته در تضاد نیست: مدل درست پاسخ نمی‌دهد، اما *به‌گونه‌ی متفاوتی* اشتباه می‌کند.

سه نکته‌ی مقایسه‌ای هم مهم است. CLPU در هر دو مجموعه‌داده پس از درخواست به $S = ۰$ می‌رسد، که پیامد ساختاری حذف کامل مدخل است و نشان می‌دهد این آماره در حالت مطلوب واقعا صفر برمی‌گرداند. مسیر PALL-Adapter و LoRA روی بدنه‌ی منجمد مقادیر بسیار کوچک‌تری دارند، که با کوچک‌تربودن دامنه‌ی به‌روزرسانی آن‌ها سازگار است. و سرانجام تأکید می‌شود که S تنها در یک جهت اطلاع‌بخش است: مقدار بزرگ نشان‌دهنده‌ی تفکیک‌پذیری واقعی است، اما مقدار کوچک هیچ تضمینی نمی‌دهد، چون ممکن است صرفا حمله‌ی به‌کاررفته ضعیف باشد. بنابراین این جدول شاهدی

علیه خوانش «حذف انجام شد» فراهم می‌کند و شاهدی به سود آن نمی‌سازد.

۴-۱۱-۳ نسبت هنجار گرادیان

معیار آخر این بخش، شدت *تداخل* را در لحظه‌ی درخواست می‌سنجد و نه کیفیت فراموشی را: نسبت هنجار ℓ_2 گرادیان وظیفه‌ی هدف به گرادیان وظایف نگه‌داری‌شده، هر دو روی زیرشبکه‌ی مشترک و پیش از بازنشانی. جدول ۴-۱۷ این نسبت را می‌دهد.

جدول ۴-۱۷: نسبت هنجار ℓ_2 گرادیان فراموشی به گرادیان نگه‌داری روی زیرشبکه‌ی مشترک، در لحظه‌ی درخواست و پیش از بازنشانی (پروتکل استاندارد، هشت بذر). مقدار نزدیک به یک یعنی دو سیگنال هم‌مرتبه‌اند. این معیار در بازخورد بررسی با هدف سنجش *تداخل* پیشنهاد شده بود و گزارش می‌شود، اما همان هدف را برآورده نمی‌کند: نسبت هنجار تنها نسبت *بزرگی* است و هیچ اطلاعاتی از جهت ندارد، پس تعارض یا هم‌راستایی دو گرادیان را نمی‌سنجد؛ برای آن به کسینوس، ضرب داخلی یا درصد علامت‌های مخالف نیاز است.

مجموعه‌داده	روش	بذر	نسبت هنجار گرادیان
CIFAR-10	PALL	8	0.7986 ± 0.3257
	EPALL	8	0.7432 ± 0.6284
CIFAR-100	PALL	8	0.6715 ± 0.2313
	EPALL	8	0.7663 ± 0.3624

مقادیر در بازه‌ی ۰/۶۷ تا ۰/۸۰ قرار می‌گیرند، یعنی دو سیگنال هم‌مرتبه هستند: گرادیان فراموشی نه آن‌قدر بزرگ است که گرادیان نگه‌داری را بی‌اثر کند و نه آن‌قدر کوچک که نادیده گرفته شود.

اما این معیار تعارض را نمی‌سنجد. چون $\|g_f\|_2/\|g_r\|_2$ تنها نسبت *بزرگی* است و هیچ اطلاعاتی از جهت ندارد، دو گرادیان می‌توانند این نسبت را برابر یک داشته باشند و در عین حال کاملا هم‌راستا یا کاملا متضاد باشند. بنابراین این جدول تنها نشان می‌دهد دو سیگنال در ناحیه‌ی مشترک *قابل مقایسه‌اند*، و ادعای «رقابت» یا «تعارض» از آن بیرون نمی‌آید. سنجش تعارض به معیارهای جهت‌دار نیاز دارد — کسینوس زاویه، ضرب داخلی، یا درصد مختصه‌هایی که علامت مخالف دارند — که هیچ‌کدام در این کار محاسبه نشده‌اند و در بخش ۵-۵ به‌عنوان کار آینده ثبت می‌شوند. تفاوت میان EPALL و PALL اصلی در این معیار کوچک و در حد انحراف معیار است، که انتظار هم همین بود: این نسبت پیش از اعمال محافظت اندازه‌گیری می‌شود و ویژگی *مسئله* است و نه ویژگی روش.

۴-۱۱-۴ مرجع بازآموزی هم‌روش

آزمون سخت‌گیرانه‌تر، مقایسه با یک نمونه‌ی تازه از همان روش و همان معماری است که از ابتدا بدون وظیفه‌ی هدف آموزش دیده باشد. جدول ۴-۱۸ نتیجه را برای دو بذر می‌دهد.

جدول ۴-۱۸: ممیزی مرجع بازآموزی هم‌روش برای EPALL. هر مرجع یک نمونه‌ی تازه از همان روش و همان معماری است که از ابتدا بدون وظیفه‌ی فراموش‌شده آموزش دیده. هر خانه به‌صورت «وظیفه‌ی فراموش‌شده / وظایف نگه‌داری‌شده» گزارش می‌شود و همه‌ی معیارهای خروجی از برش کلاس محلی وظیفه استفاده می‌کنند. این یک مقایسه‌ی تشخیصی است و اثبات فراموشی دقیق نیست.

مجموعه‌داده	بذر	توافق	JS	ℓ_2 لاجیت	کسینوس ویژگی
CIFAR-10	0	0.0920 / 0.9295	0.0219 / 0.0194	0.8111 / 1.4991	0.9085 / 0.7497
CIFAR-10	1	0.9550 / 0.9576	0.0044 / 0.0119	0.3204 / 1.4175	0.9114 / 0.7580
CIFAR-100	0	0.0000 / 0.6000	0.0769 / 0.0644	2.3804 / 1.8112	0.9211 / 0.8352
CIFAR-100	1	0.9820 / 0.5998	0.0133 / 0.0573	0.8028 / 1.7347	0.9032 / 0.8342

توافق روی وظایف نگه‌داری‌شده در بازه‌ی ۰/۶۰ تا ۰/۹۶ قرار می‌گیرد که نشان می‌دهد مدل فراموش‌کرده و مرجع بازآموزی، رفتار نسبتا مشابهی روی وظایف فعال دارند. اما توافق روی وظیفه‌ی فراموش‌شده میان ۰/۰۰ و ۰/۹۸ نوسان می‌کند. این ممیزی دوبذری *شاهدی برای هم‌ارزی با بازآموزی دقیق فراهم نمی‌کند*: مدل فراموش‌کرده و مرجع ممکن است هر دو در سطح تصادفی کار کنند اما به‌طور یکسان اشتباه نکنند. با این حال، بدون کالیبراسیون «بازآموزی در برابر بازآموزی» نمی‌توان صرفا از این نوسان، حذف دقیق توزیعی را به‌طور صوری رد کرد.

۴-۱۱-۵ ماندگاری سرکوب

درس روش‌شناختی کارهایی مانند UnCLe [۵۷] و CATA [۵۸] این بود که دقت بی‌درنگ پس از درخواست کافی نیست. جدول ۴-۱۹ وظیفه‌ی فراموش‌شده را از لحظه‌ی حذف تا پایان دنباله‌ی درخواست‌ها دنبال می‌کند. سنجه‌ی این ممیزی *فاصله تا شانس* $|A_u - c|$ است نه دقت خام: هدف حذف، رساندن وظیفه به سطح تصادفی است، پس دور شدن از شانس در هر دو جهت شکست است و افزایش دقت خام به‌تنهایی معیار درستی نیست.

نکته‌ی دوم آن است که در *دو بذر این ممیزی* (۰ و ۱)، دنباله‌ی درخواست‌ها پس از نخستین حذف *درخواست آموزشی هم دارد*؛ در بذر ۰ الگوی CIFAR-10 به‌صورت T T T T F T F F و الگوی

CIFAR-100 به‌صورت T×7 F T T F T F است. این الگو میان بذرها *یکسان نیست* و نباید به کل مجموعه تعمیم داده شود: موقعیت حذف‌ها با بذر تغییر می‌کند و در بذر ۵ مجموعه‌داده‌ی CIFAR-100 هر سه حذف در انتهای دنباله می‌آیند (T×10 F F F)، یعنی هیچ درخواست آموزشی پس از نخستین حذف وجود ندارد. بنابراین می‌توان رانش وظیفه‌ی فراموش‌شده را به دو سهم جمع‌پذیر تفکیک کرد: سهمی که از درخواست‌های آموزشی میان دو حذف می‌آید و سهمی که از خود درخواست‌های حذف بعدی می‌آید. این تفکیک از روی شناسه‌ی درخواست هر رویداد و مقادیر پیش و پس از هر حذف محاسبه می‌شود و فاصله‌ای که هیچ درخواست آموزشی در آن نباشد، به‌طور ساختاری سهم صفر می‌گیرد.

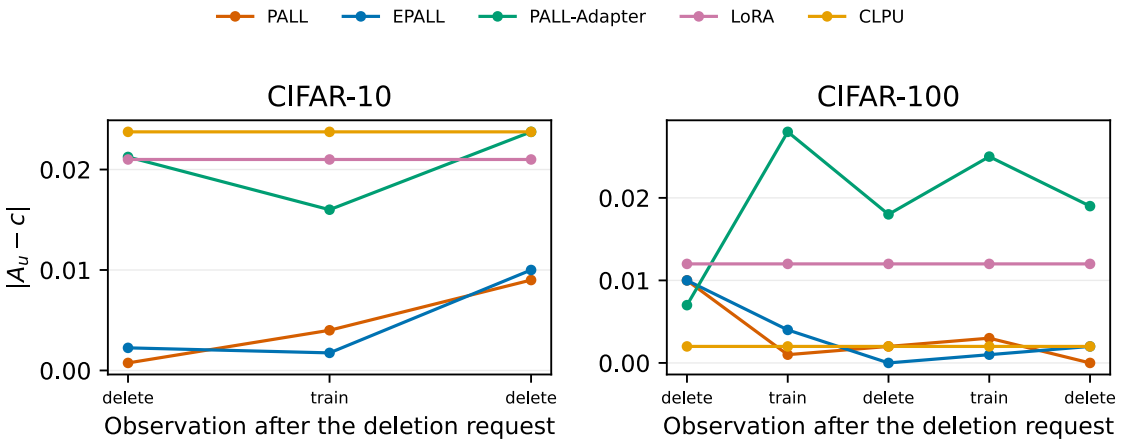
جدول ۴-۱۹: ماندگاری سرکوب وظیفه‌ی فراموش‌شده تا پایان دنباله‌ی درخواست‌ها (دو بذر، شش وظیفه‌ی حذف‌شده به‌ازای هر مجموعه‌داده و روش). همه‌ی سنجه‌ها روی *فاصله تا شانس* $|A_u - c|$ محاسبه شده‌اند نه روی دقت خام؛ برای وظیفه‌ی حذف‌شده افتادن به زیر شانس هم مانند بالا رفتن از آن، شکست فراموشی است. «بازگشت» بیشترین افزایش $|A_u - c|$ نسبت به لحظه‌ی بلافاصله پس از حذف است. در دو بذر این ممیزی (۰ و ۱) پس از نخستین حذف درخواست آموزشی هم وجود دارد — که در همه‌ی بذرها برقرار نیست — پس رانش کل به دو سهم تفکیک شده است: سهم درخواست‌های آموزشی میان‌راهی و سهم درخواست‌های حذف بعدی؛ این دو دقیقا به رانش خالص جمع می‌شوند. ستون‌های بازگشت و سهم تنها روی چهار وظیفه‌ای گزارش می‌شوند که افق پیگیری ناتهی دارند؛ وظیفه‌ی حذف‌شده در آخرین درخواست چنین افقی ندارد.

مجموعه‌داده	روش	$\lvert A_u - c\rvert$ بی‌درنگ	$\lvert A_u - c\rvert$ نهایی	میانگین بازگشت	بدترین بازگشت	سهم آموزش	سهم حذف
CIFAR-10	PALL	0.0011	0.0097	0.0179	0.0490	+0.0019	+0.0111
	EPALL	0.0018	0.0024	0.0136	0.0370	+0.0063	-0.0053
	PALL-Adapter	0.0182	0.0098	0.0024	0.0060	-0.0150	+0.0024
	LoRA	0.0071	0.0071	0.0000	0.0000	0.0000	0.0000
	CLPU	0.0198	0.0198	0.0000	0.0000	0.0000	0.0000
CIFAR-100	PALL	0.0087	0.0017	0.0030	0.0100	-0.0090	-0.0015
	EPALL	0.0137	0.0040	0.0040	0.0100	-0.0120	-0.0025
	PALL-Adapter	0.0230	0.0260	0.0145	0.0500	+0.0100	-0.0055
	LoRA	0.0053	0.0053	0.0000	0.0000	0.0000	0.0000
	CLPU	0.0140	0.0140	0.0000	0.0000	0.0000	0.0000

بدترین بازگشت مشاهده‌شده روی CIFAR-10 برابر ۰/۰۴۹۰ برای PALL و ۰/۰۳۷۰ برای EPALL و ۰/۰۰۶۰ برای PALL-Adapter است؛ روی CIFAR-100 برابر ۰/۰۵۰۰ برای PALL-Adapter و ۰/۰۱۰۰ برای PALL و EPALL. LoRA و CLPU بازگشت دقیقا صفر دارند، چون پارامترهای وظیفه‌ی حذف‌شده در آن‌ها از پارامترهای وظایف بعدی جدا هستند و هیچ درخواست بعدی آن‌ها را لمس نمی‌کند.

تفکیک سهم‌ها نتیجه‌ی غیرمنتظره‌ای می‌دهد. روی CIFAR-100، سهم درخواست‌های آموزشی برای

هر دو نسخه‌ی PALL منفی است (۰/۰۰۹۰− و ۰/۰۱۲۰−)؛ یعنی یادگیری وظایف تازه، وظیفه‌ی فراموش‌شده را به شانس نزدیک‌تر می‌کند نه دورتر. روی CIFAR-10 همین سهم اندکی مثبت است (۰/۰۰۱۹+ و ۰/۰۰۶۳+) اما بخش عمده‌ی رانش PALL از درخواست‌های حذف بعدی می‌آید (۰/۰۱۱۱+)، در حالی که همین سهم برای EPALL منفی است (۰/۰۰۵۳−). خوانش این تفاوت آن است که حفاظت رتبه‌بندی‌شده، علاوه بر مهار آسیب به وظایف نگه‌داری‌شده، مانع از آن می‌شود که حذف‌های بعدی وظیفه‌ی پیش‌تر حذف‌شده را دوباره از سطح تصادفی دور کنند. با این حال هیچ‌کدام از این تفاوت‌ها با دو بذر و چهار وظیفه‌ی دارای افق پیگیری، آزمون معناداری را برنمی‌تابد؛ آن‌ها را باید مشاهده‌ی توصیفی خواند. جمع‌بندی این است که سرکوب ماندگار است اما ناوردا نیست، و منبع غالب ناوردایی درخواست‌های حذف بعدی است نه یادگیری تازه. شکل ۴–۸ مسیرها را نشان می‌دهد.



شکل ۴–۸: مسیر فاصله تا شانس $|A_u - c|$ برای وظیفه‌ی فراموش‌شده، از لحظه‌ی حذف به بعد. منحصرا نخستین وظیفه‌ی حذف‌شده‌ی هر اجرا دنبال می‌شود. برچسب هر نقطه‌ی محور افقی نشان می‌دهد که تغییر تا آن نقطه از درخواست‌های آموزشی میان‌راهی آمده است یا از یک درخواست حذف بعدی. چون موقعیت درخواست‌های حذف با بذر تغییر می‌کند، طول و ترکیب افق پیگیری میان بذرها یکسان نیست؛ منحنی‌ها تا آخرین نقطه‌ای رسم شده‌اند که همه‌ی اجراهای آن مجموعه‌داده در آن هم حاضرند و هم نوع رویداد یکسانی دارند، و پس از آن قطع می‌شوند تا میانگین‌گیری روی رویدادهای ناهم‌جنس رخ ندهد. به همین دلیل CIFAR-10 سه نقطه و CIFAR-100 پنج نقطه دارد. هر نقطه‌ی هر منحنی میانگین دو بذر همان روش است. جدول ۴–۱۹ از این محدودیت مصون است، چون آماره‌های آن به‌ازای هر اجرا و مستقل از شکل دنباله محاسبه می‌شوند.

۴-۱۲ مطالعات حساسیت

۴-۱۲-۱ انتخاب لنگر

بخش ۳-۳-۳ اشاره کرد که لنگرزدن به θ^{old} یک ایراد مفهومی دارد: آن وزن‌ها خودشان اطلاعاتی از وظیفه‌ی فراموش‌شونده حمل می‌کنند. جدول ۴-۲۰ جایگزین θ^{reinit} را روی هشت بذر هم‌تراز از پروتکل استاندارد می‌سنجد. چون `protect_anchor` فقط روی EPALL اثر دارد، دو بازو تنها در همین یک تنظیم تفاوت دارند و بذرها جفت می‌شوند؛ بازوی `reinit` زیر برچسب آزمایش مستقل `*_standard_reinit_v1` اجرا شده تا جدول‌های اصلی دست‌نخورده بمانند.

جدول ۴-۲۰: مطالعه‌ی جمله‌ی لنگر ℓ_2 رابطه‌ی (۳-۱۱) روی هشت بذر هم‌تراز از پروتکل استاندارد. گزینه‌ی `old` به وزن‌های پیش از درخواست لنگر می‌زند و `reinit` به نمونه‌ای تازه از مقدارگذاری اولیه. دو بازو تنها در همین یک تنظیم تفاوت دارند و به‌ازای هر بذر دنباله‌ی درخواست یکسانی می‌بینند، پس هر بذر یک جفت هم‌تراز می‌سازد. ستون آخر تعداد بذرهای به سود `old` از هشت بذر است، به ترتیب برای دقت نهایی، فراموشی، بیشترین افت و دقت هدف؛ A_u بر حسب فاصله تا شانس امتیازدهی شده و مجموع کمتر از هشت به معنای وجود گره است. هیچ‌یک از ده آزمون این خانواده پس از اصلاح Holm در سطح ۰/۰۵ باقی نمی‌ماند، پس این جدول توصیفی است.

مجموعه‌داده	لنگر	دقت نهایی	فراموشی	بیشترین افت	دقت هدف	جفت به سود `old`
CIFAR-10	`old`	0.9433 ± 0.0160	0.2872 ± 0.0249	0.0027 ± 0.0031	0.4863 ± 0.0216	3 / 3 / 3 / 3
	`reinit`	0.9446 ± 0.0199	0.2787 ± 0.0125	0.0009 ± 0.0015	0.4923 ± 0.0147	
CIFAR-100	`old`	0.7371 ± 0.0093	0.1958 ± 0.0121	0.0056 ± 0.0062	0.0959 ± 0.0075	6 / 7 / 6 / 2
	`reinit`	0.7340 ± 0.0126	0.1987 ± 0.0120	0.0145 ± 0.0105	0.1001 ± 0.0037	

نتیجه‌ی هشت‌بذری با مطالعه‌ی دوبذری پیشین هم‌خوان نیست و جهت آن روی دو مجموعه‌داده یکسان نیست. روی CIFAR-100 برتری با θ^{old} است: فراموشی در هفت جفت از هشت جفت به سود آن است ($d_z = ۱/۰۶$)، بیشترین افت در شش جفت ($d_z = ۱/۰۴$)، میانگین اختلاف ۰/۰۰۸۹ و دقت نهایی در شش جفت. روی CIFAR-10 تفاوت‌ها در حد نوسان بذرهاست و اندکی به سود θ^{reinit} می‌چرخد، اما هیچ معیاری بیش از پنج جفت از هشت جفت را نمی‌گیرد و بزرگی اختلاف دقت ۰/۰۰۱۳ است. تنها معیاری که در هر دو مجموعه‌داده به سود θ^{reinit} می‌ماند، فاصله‌ی دقت هدف تا شانس است و آن هم با اختلاف میانگین ۰/۰۰۳۰ روی CIFAR-100. مهم‌تر از همه، این خانواده ده آزمون دارد و پس از اصلاح Holm هیچ معیاری در هیچ جهتی در سطح ۰/۰۵ باقی نمی‌ماند؛ کوچک‌ترین p خام برابر ۰/۰۳۵ است

که پس از اصلاح به ۰/۳۵ می‌رسد.

پیش‌فرض پیاده‌سازی θ^{old} است و همه‌ی جدول‌های این پایان‌نامه با همان تولید شده‌اند. سه دلیل پشت این انتخاب است. نخست، لنگر تنها روی $S_{\text{crit}} \subseteq M_f \cap M_r$ عمل می‌کند و فراموشی از راه بازنشانی ناحیه‌ی انحصاری $M_f \setminus M_r$ رخ می‌دهد؛ پس لنگر یک اهرم *نگه‌داری* است نه اهرم فراموشی، و باید عمدتا با معیارهای نگه‌داری داوری شود — جایی که روی CIFAR-100 برتری با θ^{old} است و روی CIFAR-10 تفاوتی نیست. دوم، θ^{reinit} آن استقلال از داده‌ای را که وعده می‌دهد در عمل نمی‌خرد: شاخه‌های بازنشانی ارثی همچنان به پشتیبان‌های پراکنده‌ی وابسته به داده‌ی قدیمی تکیه دارند، پس مسیر نشت از راه دیگری باز می‌ماند و محدودیت دامنه‌ی ادعای حذف (بخش ۵-۴) همچنان برقرار است. سوم، θ^{old} در زمان درخواست نیز سریع‌تر است (۳/۸۷ ثانیه روی CIFAR-100 و ۰/۴۷ ثانیه روی CIFAR-10 به‌طور میانگین). ایراد مفهومی θ^{old} سر جای خود باقی است و در بخش ۵-۴ به‌عنوان محدودیت ثبت می‌شود، اما شاهد هشت‌بذری آن را به توصیه‌ی عملی تبدیل نمی‌کند.

تصریح می‌شود که θ^{reinit} در بازخورد بررسی‌ای که این کار دریافت کرد، به‌عنوان صورت‌بندی درست جمله‌ی لنگر پیشنهاد شده بود؛ استدلال آن بازخورد این بود که توزیع مقدارگذاری اولیه از هر داده‌ای مستقل است، پس لنگر، اطلاعات وظیفه‌ی حذف‌شونده را دوباره وارد نمی‌کند. آن استدلال *در سطح خودِ جمله‌ی لنگر* درست است و اینجا رد نمی‌شود. آنچه گزارش می‌شود دو چیز دیگر است. نخست، در سطح *سامانه* کافی نیست: مسیر بازنشانی شاخه‌محور که از PALL به ارث رسیده همچنان از پشتیبان‌های پراکنده‌ی وابسته به داده‌ی پیشین می‌خواند، پس مستقل‌کردن لنگر تنها یکی از دو مسیر وابستگی را می‌بندد و ادعای استقلال از داده در سطح کل درخواست برقرار نمی‌شود. دوم، همان صورت‌بندی، حفظ تضمین حذف دقیق را *مشروط* بیان می‌کند: مشروط بر آنکه جمله‌ی حذف روی پارامترهای غیرمشترک به حذف دقیق برسد. ممیزی مرجع بازآموزی در بخش ۴-۱۱-۴ برای برقراری این شرط شاهدی به‌دست نمی‌دهد؛ توافق روی وظیفه‌ی هدف میان ۰/۰۰ و ۰/۹۸ نوسان می‌کند که با خوانش حذف دقیق سازگار نیست. بنابراین تغییر لنگر، مقدم آن گزاره‌ی شرطی را برقرار نمی‌کند و تالی آن نیز به‌دست نمی‌آید. در کنار این دو ملاحظه‌ی مفهومی، مقایسه‌ی هشت‌بذری بالا هیچ برتری تجربی‌ای برای θ^{reinit} نشان نداد. به همین دلیل پیشنهاد بازخورد آزموده شد اما پذیرفته نشد، و دلیل نپذیرفتن آن در همین بخش مستند است.

۴-۱۲-۲ معیار اهمیت: بزرگی گرادیان در برابر انرژی تعارض

جدول ۴-۲۱ دو معیار رتبه‌بندی معرفی‌شده در معادلات (۳-۸) و (۳-۱۰) را مقایسه می‌کند.

جدول ۴-۲۱: مقایسه‌ی دو معیار رتبه‌بندی اهمیت روی دو بذر هم‌تراز؛ توصیفی. بزرگی گرادیان رابطه‌ی (۳-۸) پیش‌فرض است و انرژی تعارض گزینه‌ی جانشین رابطه‌ی (۳-۱۰).

مجموعه‌داده	روش	معیار اهمیت	دقت نهایی	بیشترین افت
CIFAR-10	PALL-Adapter	بزرگی گرادیان	0.8166 ± 0.0249	0.0160 ± 0.0057
		انرژی تعارض	0.8259 ± 0.0136	0.0063 ± 0.0110
CIFAR-100	EPALL	بزرگی گرادیان	0.4933 ± 0.0083	0.0060 ± 0.0028
		انرژی تعارض	0.5040 ± 0.0085	0.0230 ± 0.0099

نتیجه وابسته به مسیر و مجموعه‌داده است: انرژی تعارض به مسیر آداپتر روی CIFAR-10 کمک می‌کند (بیشترین افت از ۰/۰۱۶۰ به ۰/۰۰۶۳) اما به EPALL روی CIFAR-100 آسیب می‌زند (بیشترین افت از ۰/۰۰۶۰ به ۰/۰۲۳۰). به همین دلیل بزرگی گرادیان معیار پیش‌فرض می‌ماند. نکته‌ی عملی مهم این است که معیار تعارض به گرادیان وظیفه‌ی هدف نیاز دارد و بنابراین باید در پنجره‌ی دسترسی پیش از حذف محاسبه شود؛ یعنی حتی اگر سود می‌داشت، هزینه‌ی معماری بیشتری داشت.

۴-۱۲-۳ انتقال میان‌معماری با ViT

جدول ۴-۲۲ همان مقایسه‌ی EPALL و PALL اصلی را روی یک بدنه‌ی ViT-T/8 [۳۰] تکرار می‌کند.

جدول ۴-۲۲: انتقال میان‌معماری روی بدنه‌ی ViT-T/8 با دو بذر هم‌تراز؛ توصیفی و بدون ادعای معناداری.

مجموعه‌داده	روش	دقت نهایی	بیشترین افت	دقت هدف
CIFAR-10	PALL	0.8346 ± 0.0366	0.0192 ± 0.0159	0.5135 ± 0.1846
	EPALL	0.8364 ± 0.0592	0.0193 ± 0.0230	0.5585 ± 0.0778
CIFAR-100	PALL	0.3921 ± 0.0010	0.0190 ± 0.0071	0.2160 ± 0.0792
	EPALL	0.3990 ± 0.0063	0.0100 ± 0.0028	0.2120 ± 0.0764

روی Split-CIFAR-100، EPALL بیشترین افت میانگین را از ۰/۰۱۹۰ به ۰/۰۱۰۰ کاهش می‌دهد و دقت را کمی بهبود می‌دهد (۰/۳۹۹۰ در برابر ۰/۳۹۲۱)، در حالی که A_u نزدیک به سطح تصادفی می‌ماند. روی Split-CIFAR-10 میانگین‌ها تقریبا برابرند. بنابراین انتقال به معماری ترنسفورمری وابسته به مجموعه‌داده است و با دو بذر هیچ ادعای معناداری مطرح نمی‌شود.

۴-۱۲-۴ عرض گلوگاه آداپتر

جدول ۴-۲۳ جاروب عرض گلوگاه را نشان می‌دهد. نتیجه غیریکنواخت است: نه دقت و نه بیشترین افت با افزایش عرض روند مشخصی ندارند. بنابراین انتخاب ۱۶ یک پیش‌فرض فشرده است و نه یک بهینه‌ی ادعاشده.

جدول ۴-۲۳: جاروب عرض گلوگاه مخصوص وظیفه روی CIFAR-10 با دو بذر هم‌تراز؛ توصیفی. گلوگاه مشترک روی ۱۶ ثابت است. هیچ‌یک از معیارها نسبت به عرض یکنوا نیست، پس ۱۶ یک پیش‌فرض فشرده است و نه بهینه‌ی ادعاشده.

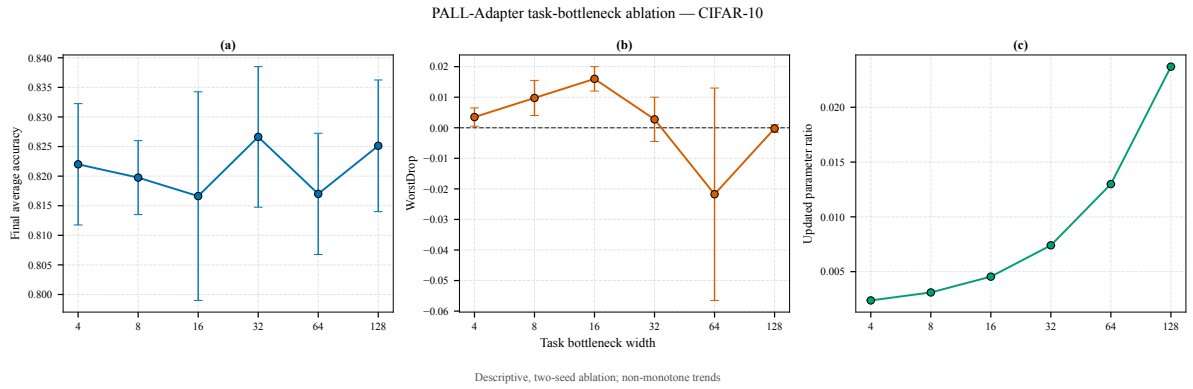
گلوگاه وظیفه	دقت نهایی	بیشترین افت	دقت هدف	R_{upd}
4	0.8220	0.0035	0.5215	0.0024
8	0.8197	0.0097	0.5425	0.0031
16	0.8166	0.0160	0.5247	0.0045
32	0.8266	0.0028	0.5058	0.0074
64	0.8170	−0.0218	0.5085	0.0130
128	0.8251	−0.0003	0.5000	0.0237

شکل‌های ۴-۹ و ۴-۱۰ نمای گرافیکی این جاروب و نقشه‌ی حرارتی دقت یک اجرای نماینده را نشان می‌دهند. علاوه بر این، جاروب ابرپارامتری پیش‌آموزش‌دیده‌ی مبتنی بر نرمال‌سازی قدیمی از شواهد تصحیح‌شده *کنار گذاشته شده* است، چون پیش‌پردازش ورودی آن با بدنه‌ی ImageNet هم‌خوان نبود.

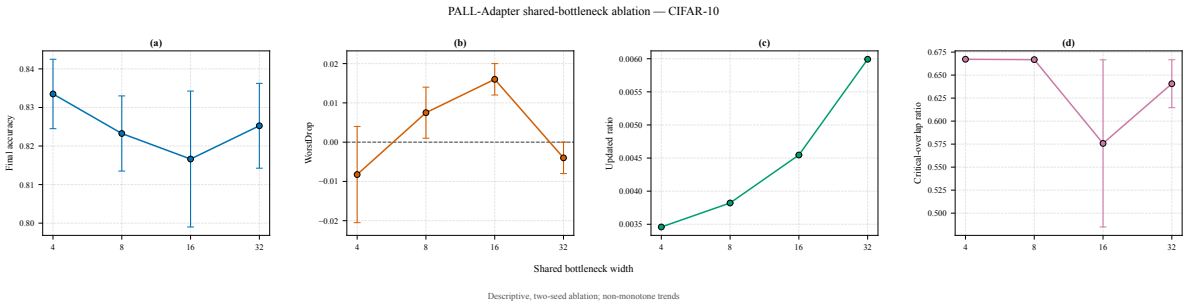
ترکیب هم‌پوشانی. لاگ‌های مسیر آداپتر نسبت $|S_{\text{crit}}|/|S_{\text{forget}}|$ را در بازه‌ی ۰٫۳۵ تا ۰٫۵۶ و کسر سفت‌محافظت‌شده را در بازه‌ی ۰٫۰۷ تا ۰٫۱۱ ثبت می‌کنند. این مقادیر نشان می‌دهند ناحیه‌ی بحرانی در آداپتر مشترک بخش قابل‌توجهی از ناحیه‌ی هدف است — یعنی شرایط برای اثرگذاری ماسک نرم فراهم بود — و بی‌اثربودن آن در نقطه‌ی پایان (بخش ۴-۸) به کوچک‌بودن ناحیه ربطی ندارد.

۴-۱۳ تهدیدهای اعتبار

اعتبار درونی. خطر اصلی، انتساب بهبود به مؤلفه‌ی نادرست است. کاهش این خطر با سه سازوکار انجام شده است: جفت‌شدگی دنباله‌ها بر اساس بذر، کنترل‌های مؤلفه‌ای که هر بار فقط یک جزء را عوض

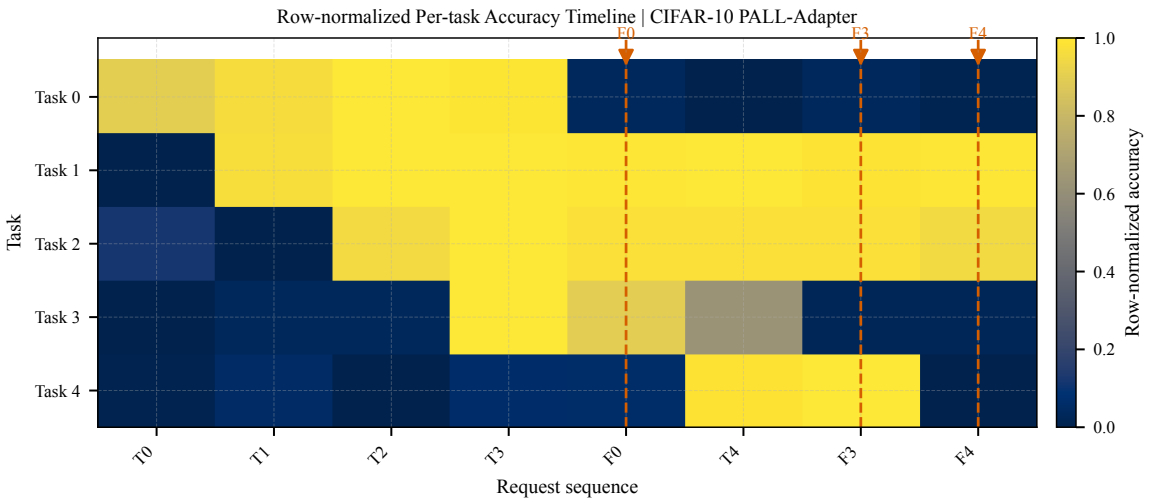


(آ) گلوگاه مخصوص وظیفه



(ب) گلوگاه مشترک

شکل ۴-۹: جاروب عرض گلوگاه مخصوص وظیفه و گلوگاه مشترک روی CIFAR-10. هیچ‌یک از دو منحنی یکنوا نیست.



شکل ۴-۱۰: نقشه‌ی حرارتی دقت وظیفه‌ها در طول یک اجرای نماینده‌ی PALL-Adapter. سطرها وظیفه‌ها و ستون‌ها درخواست‌ها هستند؛ افت ستونی در نقطه‌ی هر درخواست فراموشی دیده می‌شود.

می‌کنند، و هم‌ترازی مولد اعداد تصادفی در بازوهای آداپتر که باعث شد بی‌اثربودن ماسک نرم از نوفه قابل تفکیک باشد.

اعتبار آماری. کف p قابل‌دستیابی برای هر آزمون جداگانه است و به تعداد جفت‌های بی‌گره بستگی دارد؛ برای سطرهای بدون گره‌ی CIFAR-100 برابر ۰/۰۰۳۹۰۶ و برای دقت نهایی CIFAR-10 برابر ۰/۰۰۷۸۱۲ است. شش آزمون هم‌زمان با اصلاح Holm تصحیح شده‌اند و چهار مورد از آن‌ها در سطح ۰/۰۵ باقی می‌مانند. تحلیل‌های دو و سه بذری صریحا توصیفی برچسب خورده‌اند و در هیچ ادعای معناداری وارد نشده‌اند. بازه‌های بوت‌استرپ نیز توصیفی‌اند و آزمون فرض نیستند.

ناپایداری نتیجه با افزودن بذر. یک هشدار روش‌شناختی مشخص از همین کار بیرون آمد: افزودن سه بذر به کنترل‌های مؤلفه‌ای، جدایی جمله‌ی لنگر روی CIFAR-100 را از میان برد (بخش ۴-۶). این نشان می‌دهد بازه‌های بوت‌استرپ نمونه‌ی کوچک تا چه اندازه می‌توانند شکننده باشند و چرا انتساب سازوکار باید با کم‌ترین تعداد بذری که نتیجه در آن پایدار است گزارش نشود.

اعتبار بیرونی. همه‌ی نتایج روی سه محک تصویری با وظیفه‌های مصنوعی افرازشده به‌دست آمده‌اند. تعمیم به داده‌ی واقعی با مرزهای وظیفه‌ی نامشخص، به داده‌ی چندحالتی، یا به مقیاس‌های بزرگ‌تر آزموده نشده است. آزمون ViT تنها یک نشانه‌ی ابتدایی از انتقال میان‌معماری می‌دهد.

اعتبار سازه‌ای. خطر جدی‌تر، گرفتن معیارها به‌جای مفهوم است. سه نمونه‌ی مشخص: A_u نزدیک به سطح تصادفی معیار حذف نیست، R_{upd} معیار کارایی نیست، و AUC نزدیک به ۰/۵ در MIA معیار حریم خصوصی نیست. تمام متن این فصل بر همین تفکیک‌ها اصرار کرده است.

اعتبار محاسباتی. زمان درخواست نرمال‌نشده است و روی یک پیکربندی سخت‌افزاری خاص اندازه‌گیری شده؛ بنابراین برای مقایسه‌ی مرتبه‌ای معتبر است و نه برای مقایسه‌ی دقیق.

۴-۱۴ جمع‌بندی یافته‌ها

پاسخ‌های تجربی به پرسش‌های پژوهش چنین است.

RQ1 — بله، با دامنه‌ی مشخص. EPALL هم‌زمان A_u را به سطح تصادفی می‌رساند و دقت نگه‌داری را نسبت به PALL اصلی بهبود می‌دهد؛ این بهبود روی هر دو مجموعه‌داده پس از اصلاح Holm باقی می‌ماند. بیشترین افت تنها روی CIFAR-100 معنادار می‌ماند.

RQ2 — عملا هم‌تراز در دقت، با هزینه‌ی زمان درخواست. EPALL روی CIFAR-10 کمتر از یک واحد درصد پایین‌تر از CLPU است و روی CIFAR-100 در حد پراکندگی بذرها با آن برابر می‌ایستد، در حالی که CLPU میان ۱/۶ تا ۲/۲ برابر حالت مقیم مصرف می‌کند. در مقابل، EPALL در زمان درخواست از PALL اصلی کندتر است.

RQ3 — شواهد مثبت اما مشروط؛ اثر علّی خالص اثبات نشده است. محک موقعیت درخواست شاهدی توصیفی می‌دهد: بیشترین افت PALL اصلی با رشد درجه بالا می‌رود و EPALL مسطح می‌ماند، اما درجه‌ها هم‌پوشانی واقعی را کنترل نمی‌کنند. جاروب پراکندگی (بخش ۴-۵) با ثابت‌نگه‌داشتن هویت و موقعیت وظیفه‌ی هدف، هم‌پوشانی اندازه‌گیری‌شده را در بازه‌ای بیش از ده‌برابری جابه‌جا می‌کند و همبستگی رتبه‌ای مثبت و معناداری میان آن و سود حفاظت نشان می‌دهد؛ اما چون همان دستگیره ظرفیت را نیز تغییر می‌دهد، این شاهد در سطح اثر متقابل باقی می‌ماند و اثر علّی خالص هم‌پوشانی را جدا نمی‌کند.

RQ4 — تنها رتبه‌بندی با بودجه‌ی ثابت. کیفیت انتخاب به‌روشنی و روی هر دو مجموعه‌داده مؤثر است. دو جزء دیگر سهم مستقل نشان نمی‌دهند، اما کنترل‌های موجود برای رد ضرورت آن‌ها کافی نیستند. افزون بر این، رتبه‌بندی همه‌ی معیارها را غالب نمی‌کند: محافظت بدون رتبه‌بندی بیشترین افت کمتری دارد.

RQ5 — نه. ماسک نرم در نقطه‌ی پایان مسیر آداپتر اثر قابل‌اندازه‌گیری ندارد و سرکوب به بازنشانی مسیر هدف نسبت داده می‌شود.

فصل ۵

جمع‌بندی و کارهای آینده

۵-۱ خلاصه‌ی کار انجام‌شده

این پایان‌نامه مسئله‌ی فراموشی انتخابی در یادگیری پیوسته را از زاویه‌ی هم‌پوشانی پارامتری بررسی کرد. نقطه‌ی شروع، این مشاهده بود که آسیب جانبی یک درخواست حذف، تصادفی و غیرقابل توضیح نیست: بسط مرتبه‌ی اول بخش ۳-۲ نشان داد که بخش عمده‌ی این آسیب از ناحیه‌ی مشترک میان زیرشبکه‌ی وظیفه‌ی هدف و زیرشبکه‌های فعال انتظار می‌رود، و درون همان ناحیه هم توزیع یکنواخت نیست؛ دنباله‌ی بالای توزیع حساسیت، بخش غالب مجموع را می‌سازد.

از این تحلیل یک تجویز طراحی مشخص بیرون آمد و به روش EPALL تبدیل شد: هم‌پوشانی ساختاری $S_{\text{share}} = M_f \cap M_r$ را از ماسک‌های دودویی ذخیره‌شده به‌طور دقیق محاسبه کن، مختصه‌های آن را با بزرگی گرادیان وظایف نگه‌داری‌شده رتبه‌بندی کن، زیرمجموعه‌ی بالای ρ را انتخاب کن، و تنها همان زیرمجموعه را در طول ترمیم با یک جمله‌ی لنگر ℓ_2 مهار کن. منطق بازنشانی شاخه‌محور PALL دست‌نخورده به ارث می‌رسد و نوآوری تنها در لایه‌ی انتخاب و مهار است.

در کنار این مسیر اصلی، یک مسیر دوم — PALL-Adapter — با انگیزه‌ی کاهش دامنه‌ی به‌روزرسانی طراحی و ممیزی شد: بدنه منجمد، یک آداپتر مشترک و یک آداپتر مخصوص هر وظیفه، و یک فراموشی سه‌فازی با یک ماسک نرم روی مختصه‌های مشترک. برای این مسیر یک لم با دامنه‌ی دقیقا محدود اثبات شد (لم ۳-۴) که یک *کران بالای* افت زیرفاز می‌دهد و نشان می‌دهد آن کران نسبت به شدت محافظت p ناافزاینده است. دو تصریح بلافاصله پس از آن آمد: نخست اینکه از مقایسه‌ی دو کران بالا نمی‌توان رابطه‌ای میان دو افت *واقعی* نتیجه گرفت (مشاهده‌ی ۳-۶)، و دوم اینکه این کران به افت انتهابه‌انتها تعمیم نمی‌یابد (بخش ۳-۵-۴).

ارزیابی روی سه محک و در سه پروتکل مجزا انجام شد، با یک مقایسه میان یازده روش و روی هشت بذر، یک محک موقعیت درخواست با ۳۰۰ اجرا، دو مجموعه کنترل مؤلفه‌ای هشت‌بذری و سه‌بذری، حسابداری مستقیم حالت مقیم، و سه ممیزی مستقل حریم خصوصی و ماندگاری.

۵-۲ پاسخ به پرسش‌های پژوهش

RQ1: آیا محافظت از هم‌پوشانی حساس، هم‌زمان سرکوب و نگه‌داری را ممکن می‌کند؟ بله. EPALL وظیفه‌ی هدف را به سطح تصادفی می‌رساند و هم‌زمان دقت نگه‌داری را نسبت به PALL بهبود می‌دهد. اختلاف جفتی دقت روی هر دو مجموعه‌داده به $p = ۰/۰۰۷۸۱۲$ می‌رسد و پس از اصلاح Holm نیز معنادار می‌ماند؛ بیشترین افت در هر دو بهبود می‌یابد اما تنها روی CIFAR-100 از اصلاح چندگانگی عبور می‌کند. تنها معیاری که بهبود پیوسته نشان نمی‌دهد F_{avg} روی CIFAR-10 است. جزئیات در بخش ۴-۳ و جدول‌های آن.

RQ2: یک روش مدل‌مشترک چقدر به مرجع جداسازی نزدیک می‌شود؟ بسیار نزدیک، اما این نزدیکی هم‌ارزی آماری *اثبات‌شده* نیست، چون هیچ آزمون هم‌ارزی برای این جفت انجام نشده است. روی CIFAR-10 فاصله‌ی دقت از CLPU کمتر از یک واحد درصد است و روی CIFAR-100 دو مقدار عددا نزدیک‌اند. هزینه‌ی این نزدیکی در زمان درخواست پرداخت می‌شود و مزیت آن در حالت مقیم به‌دست می‌آید: CLPU میان ۱/۶ تا ۲/۲ برابر حالت حسابداری‌شده مصرف می‌کند و سریع‌تر رشد می‌کند. نتیجه‌ی عملی: وقتی نگه‌داشتن یک شبکه‌ی کامل به‌ازای هر وظیفه مقدور باشد CLPU انتخاب بهتری است، و وقتی نباشد EPALL تقریبا همان نقطه را با یک مدل مشترک به‌دست می‌آورد.

RQ3: آیا محافظت با رشد هم‌پوشانی مهم‌تر می‌شود؟ شواهد مثبت اما مشروط است؛ اثر علّی خالص اثبات نشده است. بخش ۴-۵ دنباله، وظیفه‌ی هدف و موقعیت آن را ثابت نگه می‌دارد و پراکندگی زیرشبکه را تغییر می‌دهد؛ در نتیجه $|S_{\text{share}}|$ در بازه‌ای بیش از ده‌برابری جابه‌جا می‌شود و سود حفاظت با آن همبستگی رتبه‌ای مثبت و معنادار دارد. این رابطه روی CIFAR-10 یکنوا و روی CIFAR-100 مثبت اما غیریکنواست. چون همان دستگیره ظرفیت را هم تغییر می‌دهد و آزمون‌های درون‌خانه‌ای پنج‌بذری ضعیف‌اند، نتیجه پشتیبان فرضیه است نه شناسایی علّی مستقل هم‌پوشانی. محک موقعیت درخواست (بخش ۴-۴) نیز با همین جهت سازگار است، اما تنها شاهد توصیفی است.

RQ4: کدام مؤلفه منشأ بهبود است؟ تنها *کیفیت انتخاب*. بازوی بودجه‌ی تصادفی با همان تعداد مختصه در هر دو مجموعه‌داده بدتر است و بازه‌ی آن صفر را در بر نمی‌گیرد، پس رتبه‌بندی حساسیت و نه صرفا اندازه‌ی بودجه مؤثر است. دو مؤلفه‌ی دیگر سهم مستقل نشان نمی‌دهند و هر دو تفسیر محتاطانه می‌خواهند. بازوی رتبه‌بندی‌بدون‌اشتراک بازوی کامل را دقیقا بازتولید می‌کند، ولی این ضرورت عمومی هم‌پوشانی را رد نمی‌کند: ماسک ترمیم در هر حالت $M_f \cap M_r$ است، پس فیلتر اشتراک در پایین‌دست اعمال می‌شود و آزمایش تنها می‌گوید اعمال دوباره‌ی آن در بالادست اضافی است. بازوی بدون‌جریمه‌ی ℓ_2 نیز جدا نمی‌شود، ولی آن بازو فقط جمله‌ی جریمه را برمی‌دارد و پشتیبان‌ها را نگه می‌دارد، پس نتیجه‌اش درباره‌ی شدت مهار است و نه کل محافظت. افزون بر این، بازوی فقط‌هم‌پوشانی بیشترین افت کمتری دارد، پس رتبه‌بندی همه‌ی معیارها را غالب نمی‌کند و یک مصالحه است.

RQ5: آیا سازوکار آگاه از هم‌پوشانی در مسیر کارآمد پارامتری کار می‌کند؟ نه. بازوهای کامل و یکنواخت عملا یکسان‌اند و بازوهای کامل و ماسک/بدون‌صعود در دقت و بیشترین افت تفاوتی ندارند. چون بازوها از نظر مولد اعداد تصادفی هم‌تراز طراحی شده‌اند، این یکسانی نوفه نیست. تنها مؤلفه‌ی با اثر قابل‌اندازه‌گیری، گام صعود طبقه‌بند است. روی CIFAR-100 بازوی فقط‌بازنشانی حتی بهتر است. سرکوب مشاهده‌شده در این مسیر به بازنشانی مسیر هدف نسبت داده می‌شود و نه به ماسک نرم. جاروب شدت ماسک پیوسته در بخش ۴-۸-۱ این نتیجه را تقویت می‌کند و علت آن را نیز روشن می‌سازد: ماسک با تغییر γ واقعا جابه‌جا می‌شود و سیگنال تعارض هم وجود دارد، اما انرژی تعارض بهنجارشده در دنباله‌ای بسیار نازک متمرکز است، پس یک ضریب خطی عملا هیچ مختصه‌ای را به‌طور محسوس تضعیف نمی‌کند. مسیر پیشنهادی برای کار آینده، ماسکی مبتنی بر رتبه است و نه بر مقدار.

۵-۳ مشارکت‌های نهایی

با توجه به یافته‌های بالا، مشارکت‌های تاییدشده‌ی این پایان‌نامه چنین است:

۱. **یک سازوکار محافظت از وزن‌های مشترکِ حساس به نگه‌داری** که در تنظیم مدل‌مشترک به مرجع جداسازی کامل می‌رسد و برتری خود را هم با رشد موقعیت درخواست و هم با رشد هم‌پوشانیِ اندازه‌گیری‌شده در جاروب پراکندگی حفظ می‌کند. اجزای تاییدشده‌ی آن، رتبه‌بندی حساسیت با بودجه‌ی ثابت است.

۲. **یک ارزیابی هم‌تراز با تحلیل جفتی و آزمون دقیق** روی هشت بذر، به‌همراه دو محک مکمل برای ادعای هم‌پوشانی: یک محک موقعیت درخواست که درجه را با ساخت دنباله و از پیش تعیین

می‌کند (شاخصی جانشین و نه کنترل مستقیم هم‌پوشانی)، و یک جاروب پراکندگی که هم‌پوشانی اندازه‌گیری‌شده را مستقیما در بازه‌ای بیش از ده‌برابری جابه‌جا می‌کند اما ظرفیت را نیز هم‌زمان تغییر می‌دهد.

۳. **یک ممیزی مؤلفه‌ای دوطرفه** که برای روش اصلی، جزء مؤثر را از دو جزء بی‌اثر جدا می‌کند و برای مسیر آداپتر به یک نتیجه‌ی منفی صریح می‌رسد.

۴. **حسابداری مستقیم هزینه** که تفاوت میان نسبت دامنه‌ی ترمیم و حالت مقیم واقعی را نشان می‌دهد و از تفسیر نسبت اسمی به‌عنوان معیار کارایی جلوگیری می‌کند.

۵. **یک تحلیل نظری با دامنه‌ی صریحا محدود:** یک کران بالای جانشین برای افت زیرفاز به‌همراه اثبات یکنوایی آن نسبت به شدت محافظت، و در کنارش تصریح اینکه این کران نه رابطه‌ای میان افت‌های واقعی برقرار می‌کند و نه به نقطه‌ی پایان درخواست تعمیم می‌یابد.

۵-۴ محدودیت‌ها

محدودیت‌ها به‌صراحت و بدون تخفیف فهرست می‌شوند.

رابط و سناریو. روش به شناسه‌ی وظیفه در استنتاج نیاز دارد و سناریوی افزایشی‌کلاس را پوشش نمی‌دهد. واحد حذف، یک وظیفه‌ی کامل است و حذف در سطح رکورد یا کلاس بررسی نشده است.

دامنه‌ی ادعای حذف. تنها مدخل بازپخش درون‌فرآیندی وظیفه‌ی هدف حذف می‌شود؛ داده‌ی خام، نقاط‌بازرسی، پشتیبان‌ها و نسخه‌های بیرونی موضوع هیچ ادعایی نیستند. هیچ نتیجه‌ای حذف دقیق، گواهی‌شده یا در سطح ذخیره‌سازی را اثبات نمی‌کند. ممیزی مرجع بازآموزی (بخش ۴-۱۱-۴) با نوسان توافق میان ۰/۰۰ و ۰/۹۸ روی وظیفه‌ی هدف، شاهدی برای هم‌ارزی با بازآموزی دقیق فراهم نمی‌کند. این ممیزی برای رد صوری حذف توزیعی نیز کافی نیست: هر ردیف تنها یک جفت «مدل فراموش‌کرده در برابر یک مرجع بازآموزی» است و مرجع پایه‌ی لازم — یعنی پراکندگی توافق میان *دو مرجع بازآموزی مستقل* با بذرهای متفاوت — محاسبه نشده است. بدون آن کالیبراسیون نمی‌توان دانست چه مقدار از این نوسان صرفا نوفه‌ی آموزش است.

توان آماری و چندگانگی. کف p قابل‌دستیابی برای هر آزمون جداگانه است (۰/۰۰۳۹۰۶ برای سطرهای بی‌گره‌ی CIFAR-100 و ۰/۰۰۷۸۱۲ برای دقت نهایی CIFAR-10)، پس مقادیر مشاهده‌شده روی کف

نایستاده‌اند. شش آزمون با اصلاح Holm تصحیح شده‌اند و چهار مورد باقی می‌مانند؛ بیشترین افت CIFAR-10 پس از اصلاح از دست می‌رود. تحلیل لنگر هشت‌بذری است؛ تحلیل‌های معیار تعارض، ViT، ماندگاری و عرض گلوگاه با دو بذر انجام شده‌اند و توصیفی‌اند. یک هشدار مشخص نیز باید ثبت شود: افزودن سه بذر به کنترل‌های مؤلفه‌ای، جدایی جمله‌ی لنگر را از میان برد؛ بنابراین بازه‌های نمونه‌ی کوچک در این حوزه شکننده‌اند.

ممیزی حریم خصوصی و یک یافته‌ی نامساعد. حمله‌ی آستانه‌ی زیان نتیجه‌ی نزدیک به ۰/۵ می‌دهد که یک نتیجه‌ی تهیِ کم‌توان است. اما آماره‌ی تفکیک‌پذیری بدترین آستانه (بخش ۴-۱۱-۲) تصویر نگران‌کننده‌تری می‌دهد: S برای EPALL از صفر پیش از درخواست به میانگین حدود ۰/۱۰ و بیشینه‌ی ۰/۵۵ پس از درخواست می‌رسد، یعنی عملیات فراموشی نمونه‌های وظیفه‌ی هدف را تفکیک‌پذیرتر می‌کند. این یافته با دقت هدفِ نزدیک به سطح تصادفی در تضاد نیست، اما نشان می‌دهد سرکوب رفتاری با حذف اثر یکی نیست. تأکید می‌شود که S پارامتر حریم خصوصی تفاضلی نیست؛ ممیزی صوری و حمله‌های مبتنی بر مدل سایه [۱۱] هر دو انجام نشده‌اند و مقدار کوچک S برای هیچ روشی تضمین محسوب نمی‌شود.

ادعای سازوکاری باریک‌تر از توصیف روش. آنچه به‌طور تجربی تایید شده «رتبه‌بندی حساسیت نگه‌داری با بودجه‌ی ثابت» است. دو جزء دیگر سهم مستقل نشان ندادند، اما کنترل‌های موجود برای رد آن‌ها هم کافی نیستند: بازوی بدون‌جریمه پشتیبان‌ها را نگه می‌دارد و بازوی بدون‌اشتراک همچنان از ماسک ترمیم $M_f \cap M_r$ عبور می‌کند. بنابراین نتیجه‌ی امن این است که شواهد از رتبه‌بندی حساسیت پشتیبانی می‌کنند و درباره‌ی ضرورت سایر اجزا ساکت‌اند. این یک محدودیت واقعی و مهم در انتساب سازوکار است: روش کار می‌کند، اما نه به‌تمامی به دلیلی که در طراحی برای آن پیش‌بینی شده بود.

مسیر آداپتر بی‌اثر است. به‌رغم طراحی و تحلیل نظری، ماسک نرم در نقطه‌ی پایان اثری ندارد. این مسیر یک گزینه‌ی اکتشافی ممیزی‌شده می‌ماند و به‌عنوان راه‌حل ارائه نمی‌شود.

ماندگاری آزموده‌ی جزئی. در دو بذری که این ممیزی روی آن‌ها انجام شده، پس از نخستین حذف تنها یک تا دو درخواست آموزشی وجود دارد و افق پیگیری هر وظیفه‌ی حذف‌شده حداکثر چند درخواست است؛ این وضعیت به همه‌ی بذرها تعمیم نمی‌یابد و در دست‌کم یک بذر هیچ آموزشی پس از نخستین حذف رخ نمی‌دهد. بنابراین اگرچه ممیزی بخش ۴-۱۱-۵ سهم یادگیری تازه را از سهم حذف‌های بعدی جدا می‌کند، این افق کوتاه برای رد کردن بازگشت پس از یادگیری طولانی‌مدت — که UnCLe و CATA

روی آن تأکید دارند — کافی نیست.

هزینه‌ها. زمان درخواست نرمال‌نشده است، EPALL از PALL اصلی کندتر است، تنظیم ابرپارامترهای آداپتر خودکار نیست، و در تنظیم بیست‌وظیفه‌ای TinyImageNet بیشترین افت هر دو نسخه‌ی کامل بالا می‌ماند که به محدودیت ظرفیت اشاره می‌کند و نه فقط به هم‌پوشانی.

۵-۵ کارهای آینده

ساده‌سازی روش بر پایه‌ی یافته‌ی کنترل‌ها. دو یافته‌ی منفی کنترل‌ها یک مسیر تحقیقی مستقیم پیشنهاد می‌دهند: آیا می‌توان هم عملیات اشتراک ماسک‌ها و هم جمله‌ی لنگر ℓ_2 را کنار گذاشت و به رتبه‌بندی حساسیت با بودجه‌ی ثابت — مثلا به‌صورت صفرکردن ساده‌ی گرادیان روی مجموعه‌ی انتخاب‌شده — بسنده کرد؟ چنین نسخه‌ای هم ارزان‌تر است و هم یک ابرپارامتر (λ) کمتر دارد. آزمون این فرضیه روی پیکربندی‌های متنوع‌تر — سطوح پراکندگی مختلف، تعداد وظیفه‌ی بیشتر، و دنباله‌های با ترتیب متفاوت — می‌تواند روش را ساده‌تر کند یا شرایطی را پیدا کند که در آن آن دو جزء واقعا لازم می‌شوند.

لنگر مستقل از داده‌ی واقعی. مقایسه‌ی هشت‌بذری بخش ۴-۱۲-۱ نشان داد که جایگزینی θ^{old} با θ^{reinit} ایراد مفهومی را برطرف نمی‌کند و در عمل هم بهبودی نمی‌آورد. مسیر درست، حذف همه‌ی وابستگی‌های به داده‌ی قدیمی از مسیر درخواست است و نه فقط لنگر: تا وقتی شاخه‌های بازنشانی ارثی به پشتیبان‌های پراکنده تکیه دارند، هر لنگر مستقل از داده تنها یکی از دو مسیر نشت را می‌بندد. طراحی نسخه‌ای که هر دو مسیر را ببندد، و سنجش هزینه‌ی نگه‌داری آن، کار آینده است.

معیارهای جهت‌دار و کالیبراسیون مرجع. دو شکاف اندازه‌گیری باقی مانده است. نخست، برای پشتیبانی از ادعای «تعارض» باید معیارهای جهت‌دار گزارش شوند: کسینوس زاویه‌ی g_f و g_r، ضرب داخلی آن‌ها، و درصد مختصه‌هایی که علامت مخالف دارند. دوم، ممیزی مرجع بازآموزی باید با یک خط پایه‌ی «بازآموزی در برابر بازآموزی» کالیبره شود تا معلوم شود چه مقدار از نوسان توافق صرفا نوفه‌ی آموزش است. هر دو از لاگ‌های موجود یا با هزینه‌ی محاسباتی اندک به‌دست می‌آیند.

یک آزمایش هم‌پوشانی ظرفیت‌همتراز. جاروب پراکندگی بخش ۴-۵ هویت وظیفه‌ی هدف و ترتیب درخواست‌ها را ثابت نگه می‌دارد و دامنه‌ی گسترده‌ای در هم‌پوشانی ساختاری ایجاد می‌کند، اما همان دستگیره ظرفیت در دسترس هر وظیفه را هم تغییر می‌دهد. گام بعدی باید هم‌پوشانی را با ظرفیت

ثابت دستکاری کند — برای نمونه با افراز یا اتصال ماسک‌های ظرفیت‌همتراز — و برهم‌کنش $\mathrm{WorstDrop} \sim \mathrm{Method} \times \mathrm{MeasuredOverlap}$ را در سطح بذر بیازماید. تنها چنین طرحی می‌تواند اثر هم‌پوشانی را از اثر ظرفیت جدا کند.

ممیزی حریم خصوصی با توان بالاتر. جایگزینی حمله‌ی آستانه‌ی زیان با حمله‌های مبتنی بر مدل سایه و افزودن ممیزی صوری، ضروری‌ترین گام بعدی است. تنها با چنین ابزاری می‌توان از ادعای «سرکوب رفتاری» به ادعایی درباره‌ی اطلاعات باقی‌مانده حرکت کرد.

بودجه‌ی محافظت تطبیقی. نسبت ρ و ضریب λ در این کار ثابت‌اند. یک نسخه‌ی تطبیقی که بودجه را بر پایه‌ی هم‌پوشانی اندازه‌گیری‌شده‌ی هر درخواست تعیین کند، از نظر مفهومی جذاب است و در پیاده‌سازی نیز به‌صورت یک گزینه موجود است، اما ارزیابی هم‌تراز آن انجام نشده و می‌تواند موضوع کار بعدی باشد.

ماندگاری زیر یادگیری تازه. افزودن درخواست‌های آموزشی پس از حذف به دنباله‌ها، سنجش بازگشت پس از یادگیری را ممکن می‌کند. این پرسش برای سامانه‌های واقعی که پس از حذف به کار خود ادامه می‌دهند، عملی‌ترین پرسش باقی‌مانده است.

حذف زیر–وظیفه‌ای. گسترش واحد حذف از یک وظیفه‌ی کامل به یک کلاس یا زیرمجموعه‌ای از نمونه‌ها، هم مسئله‌ی جدیدی در تعریف ماسک ایجاد می‌کند و هم به سناریوهای واقعی حق فراموش‌شدن نزدیک‌تر است.

مقیاس و دامنه. ارزیابی روی معماری‌های بزرگ‌تر، مدل‌های چندحالتی و داده‌های واقعی با مرزهای وظیفه‌ی نامشخص، آزمون واقعی تعمیم‌پذیری این سازوکار است.

۵–۶ سخن پایانی

پیام اصلی این پایان‌نامه آن است که آسیب جانبی فراموشی در یک مدل مشترک *ساختاریافته* و با شناخت ساختار مهارشدنی است. در عین حال، طراحی معقول و تحلیل درست اثرگذاری تجربی را تضمین نمی‌کنند: مسیر آداپتر با وجود سازوکار منطقی، گزاره‌ی نظری درست و شرایط اندازه‌گیری مناسب، در نقطه‌ی پایان بی‌اثر بود. این واقعیت تنها با کنترل‌هایی آشکار شد که برای رد فرضیه طراحی شده بودند. گزارش این نتیجه‌ی منفی، در کنار نتیجه‌ی مثبت EPALL، بخشی از دستاورد این کار است.

مراجع

[1] Y. LeCun, Y. Bengio, and G. Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015.

[2] I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*. MIT Press, 2016.

[3] S. Thrun. Lifelong learning algorithms. In S. Thrun and L. Pratt, editors, *Learning to Learn*, pages 181–209. Springer, Boston, MA, 1998.

[4] G. I. Parisi, R. Kemker, J. L. Part, C. Kanan, and S. Wermter. Continual lifelong learning with neural networks: A review. *Neural Networks*, 113:54–71, 2019.

[5] M. De Lange, R. Aljundi, M. Masana, S. Parisot, X. Jia, A. Leonardis, G. Slabaugh, and T. Tuytelaars. A continual learning survey: Defying forgetting in classification tasks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(7):3366–3385, 2022.

[6] M. McCloskey and N. J. Cohen. Catastrophic interference in connectionist networks: The sequential learning problem. In *Psychology of Learning and Motivation*, volume 24, pages 109–165. Academic Press, 1989.

[7] J. Kirkpatrick, R. Pascanu, N. Rabinowitz, J. Veness, G. Desjardins, A. A. Rusu, K. Milan, J. Quan, T. Ramalho, A. Grabska-Barwinska, D. Hassabis, C. Clopath, D. Kumaran, and R. Hadsell. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, 114(13):3521–3526, 2017.

[8] European Union. Regulation (EU) 2016/679 of the European Parliament and of the Council (General Data Protection Regulation). Official Journal of the European Union, 2016.

[9] California State Legislature. California civil code title 1.81.5, the California Consumer Privacy Act of 2018 (CCPA), 2018.

[10] R. Shokri, M. Stronati, C. Song, and V. Shmatikov. Membership inference attacks against machine learning models. In *2017 IEEE Symposium on Security and Privacy (SP)*, pages 3–18. IEEE, 2017.

[11] N. Carlini, S. Chien, M. Nasr, S. Song, A. Terzis, and F. Tramèr. Membership inference attacks from first principles. In *2022 IEEE Symposium on Security and Privacy (SP)*, pages 1897–1914. IEEE, 2022.

[12] Y. Cao and J. Yang. Towards making systems forget with machine unlearning. In *2015 IEEE Symposium on Security and Privacy*, pages 463–480. IEEE, 2015.

[13] L. Bourtoule, V. Chandrasekaran, C. A. Choquette-Choo, H. Jia, A. Travers, B. Zhang, D. Lie, and N. Papernot. Machine unlearning. In *2021 IEEE Symposium on Security and Privacy (SP)*, pages 141–159. IEEE, 2021.

[14] A. Golatkar, A. Achille, and S. Soatto. Forgetting outside the box: Scrubbing deep networks of information accessible from input-output observations. In *European Conference on Computer Vision*, pages 383–398. Springer, 2020.

[15] O. Özdenizci, E. Rueckert, and R. Legenstein. Privacy-aware lifelong learning. In *International Conference on Learning Representations (ICLR)*, 2025.

[16] B. Liu, Q. Liu, and P. Stone. Continual learning and private unlearning. In *Proceedings of the 1st Conference on Lifelong Learning Agents (CoLLAs)*, pages 243–254. PMLR, 2022.

[17] C. Guo, T. Goldstein, A. Hannun, and L. van der Maaten. Certified data removal from machine learning models. In *Proceedings of the 37th International Conference on Machine Learning*, pages 3832–3842. PMLR, 2020.

[18] S. Neel, A. Roth, and S. Sharifi-Malvajerdi. Descent-to-delete: Gradient-based methods for machine unlearning. In *Proceedings of the 32nd International Conference on Algorithmic Learning Theory*, pages 931–962. PMLR, 2021.

[19] A. Sekhari, J. Acharya, G. Kamath, and A. T. Suresh. Remember what you want to forget: Algorithms for machine unlearning. In *Advances in Neural Information Processing Systems*, volume 34, 2021.

[20] T. Steinke, M. Nasr, and M. Jagielski. Privacy auditing with one (1) training run. In *Advances in Neural Information Processing Systems*, volume 36, pages 49268–49280, 2023.

[21] G. M. van de Ven, T. Tuytelaars, and A. S. Tolias. Three types of incremental learning. *Nature Machine Intelligence*, 4:1185–1197, 2022.

[22] L. Wang, X. Zhang, H. Su, and J. Zhu. A comprehensive survey of continual learning: Theory, method and application, 2023.

[23] T. T. Nguyen, T. T. Huynh, Z. Ren, P. L. Nguyen, A. W.-C. Liew, H. Yin, and Q. V. H. Nguyen. A survey of machine unlearning, 2022.

[24] H. Xu, T. Zhu, L. Zhang, W. Zhou, and P. S. Yu. Machine unlearning: A survey, 2023.

[25] H. Robbins and S. Monro. A stochastic approximation method. *The Annals of Mathematical Statistics*, 22(3):400–407, 1951.

[26] F. Zenke, B. Poole, and S. Ganguli. Continual learning through synaptic intelligence. In *Proceedings of the 34th International Conference on Machine Learning*, pages 3987–3995. PMLR, 2017.

[27] R. Aljundi, F. Babiloni, M. Elhoseiny, M. Rohrbach, and T. Tuytelaars. Memory aware synapses: Learning what (not) to forget. In *European Conference on Computer Vision*, pages 139–154, 2018.

[28] P. W. Koh and P. Liang. Understanding black-box predictions via influence functions. In *International Conference on Machine Learning (ICML)*, pages 1885–1894, 2017.

[29] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

[30] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations (ICLR)*, 2021.

[31] K. He, X. Zhang, S. Ren, and J. Sun. Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 1026–1034, 2015.

[32] Z. Li and D. Hoiem. Learning without forgetting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(12):2935–2947, 2017.

[33] G. Hinton, O. Vinyals, and J. Dean. Distilling the knowledge in a neural network. In *NIPS Deep Learning and Representation Learning Workshop*, 2015.

[34] S.-A. Rebuffi, A. Kolesnikov, G. Sperl, and C. H. Lampert. iCaRL: Incremental classifier and representation learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2001–2010, 2017.

[35] P. Buzzega, M. Boschini, A. Porrello, D. Abati, and S. Calderara. Dark experience for general continual learning: A strong, simple baseline. In *Advances in Neural Information Processing Systems*, volume 33, pages 15920–15930, 2020.

[36] H. Shin, J. K. Lee, J. Kim, and J. Kim. Continual learning with deep generative replay. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

[37] D. Lopez-Paz and M. Ranzato. Gradient episodic memory for continual learning. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

[38] A. Chaudhry, M. Ranzato, M. Rohrbach, and M. Elhoseiny. Efficient lifelong learning with A-GEM. In *International Conference on Learning Representations (ICLR)*, 2019.

[39] A. Mallya and S. Lazebnik. PackNet: Adding multiple tasks to a single network by iterative pruning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7765–7773, 2018.

[40] J. Serrà, D. Surís, M. Miron, and A. Karatzoglou. Overcoming catastrophic forgetting with hard attention to the task. In *Proceedings of the 35th International Conference on Machine Learning*, pages 4548–4557. PMLR, 2018.

[41] M. Wortsman, V. Ramanujan, R. Liu, A. Kembhavi, M. Rastegari, J. Yosinski, and A. Farhadi. Supermasks in superposition. In *Advances in Neural Information Processing Systems*, volume 33, 2020.

[42] A. A. Rusu, N. C. Rabinowitz, G. Desjardins, H. Soyer, J. Kirkpatrick, K. Kavukcuoglu, R. Pascanu, and R. Hadsell. Progressive neural networks, 2016.

[43] J. Frankle and M. Carbin. The lottery ticket hypothesis: Finding sparse, trainable neural networks. In *International Conference on Learning Representations (ICLR)*, 2019.

[44] V. Ramanujan, M. Wortsman, A. Kembhavi, A. Farhadi, and M. Rastegari. What's hidden in a randomly weighted neural network? In *Proceedings of the IEEE/CVF*

Conference on Computer Vision and Pattern Recognition (CVPR), pages 11893–11902, 2020.

[45] H. Kang, R. J. L. Mina, S. R. H. Madjid, J. Yoon, M. Hasegawa-Johnson, S. J. Hwang, and C. D. Yoo. Forget-free continual learning with winning subnetworks. In *Proceedings of the 39th International Conference on Machine Learning*, pages 10734–10750. PMLR, 2022.

[46] L. Graves, V. Nagisetty, and V. Ganesh. Amnesiac machine learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, 2021.

[47] A. Ginart, M. Y. Guan, G. Valiant, and J. Zou. Making ai forget you: Data deletion in machine learning. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

[48] Z. Izzo, M. A. Smart, K. Chaudhuri, and J. Zou. Approximate data deletion from machine learning models. In *International Conference on Artificial Intelligence and Statistics*, pages 2008–2016. PMLR, 2021.

[49] M. Chen, W. Gao, G. Liu, K. Peng, and C. Wang. Boundary unlearning: Rapid forgetting of deep networks via shifting the decision boundary. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7766–7775, 2023.

[50] V. S. Chundawat, A. K. Tarun, M. Mandal, and M. Kankanhalli. Can bad teaching induce forgetting? unlearning in deep networks using an incompetent teacher. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 7214–7222, 2023.

[51] C. Fan, J. Liu, Y. Zhang, E. Wong, D. Wei, and S. Liu. SalUn: Empowering machine unlearning via gradient-based weight saliency in both image classification and generation. In *International Conference on Learning Representations (ICLR)*, 2024.

[52] J. Foster, S. Schoepf, and A. Brintrup. Fast machine unlearning without retraining through selective synaptic dampening. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 12043–12051, 2024.

[53] C. Dwork and A. Roth. *The Algorithmic Foundations of Differential Privacy*. Now Publishers, 2014.

[54] Y. Qu, X. Yuan, M. Ding, W. Ni, T. Rakotoarivelo, and D. Smith. Learn to unlearn: A survey on machine unlearning, 2023.

[55] T. Shibata, G. Irie, D. Ikami, and Y. Mitsuzumi. Learning with selective forgetting. In Z.-H. Zhou, editor, *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*, pages 989–996. International Joint Conferences on Artificial Intelligence Organization, 8 2021. Main Track.

[56] L. Shan, W. Zhou, W. Li, and X. Ding. Lifelong learning and selective forgetting via contrastive strategy, 2024.

[57] S. Adhikari, V. Kumaravelu, and P. K. Srijith. An unlearning framework for continual learning, 2025.

[58] S. Lin, J. Dong, R. Chen, X. Zhang, L. Xu, and X. Chen. CATA: Continual machine unlearning via conflict-averse task arithmetic, 2026.

[59] R. Chatterjee, V. Chundawat, A. Tarun, A. Mali, and M. Mandal. A unified framework for continual learning and unlearning, 2024.

[60] N. Houlsby, A. Giurgiu, S. Jastrzebski, B. Morrone, Q. de Laroussilhe, A. Gesmundo, M. Attariyan, and S. Gelly. Parameter-efficient transfer learning for NLP. In *Proceedings of the 36th International Conference on Machine Learning*, pages 2790–2799. PMLR, 2019.

[61] V. Nair and G. E. Hinton. Rectified linear units improve restricted Boltzmann machines. In *Proceedings of the 27th International Conference on Machine Learning (ICML)*, pages 807–814, 2010.

[62] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations (ICLR)*, 2022.

[63] S.-A. Rebuffi, H. Bilen, and A. Vedaldi. Learning multiple visual domains with residual adapters. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

[64] J. Pfeiffer, A. Rücklé, C. Poth, A. Kamath, I. Vulić, S. Ruder, K. Cho, and I. Gurevych. Adapterhub: A framework for adapting transformers. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 46–54, 2020.

[65] R. K. Mahabadi, J. Henderson, and S. Ruder. Compacter: Efficient low-rank hypercomplex adapter layers. In *Advances in Neural Information Processing Systems*, volume 34, pages 1022–1035, 2021.

[66] A. Rücklé, G. Geigle, M. Glöckner, T. Beck, J. Pfeiffer, N. Reimers, and I. Gurevych. Adapterdrop: On the efficiency of adapters in transformers. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7930–7946, 2021.

[67] X. L. Li and P. Liang. Prefix-tuning: Optimizing continuous prompts for generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics*, pages 4582–4597, 2021.

[68] B. Lester, R. Al-Rfou, and N. Constant. The power of scale for parameter-efficient prompt tuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3045–3059, 2021.

[69] J. Rachapudi, R. Vatsi, P. Hambarde, and A. Shukla. BID-LoRA: A parameter-efficient framework for continual learning and unlearning, 2026.

[70] S. Yeom, I. Giacomelli, M. Fredrikson, and S. Jha. Privacy risk in machine learning: Analyzing the connection to overfitting. In *2018 IEEE 31st Computer Security Foundations Symposium (CSF)*, pages 268–282. IEEE, 2018.

[71] M. Nasr, R. Shokri, and A. Houmansadr. Comprehensive privacy analysis of deep learning: Passive and active white-box inference attacks against centralized and federated learning. In *2019 IEEE Symposium on Security and Privacy*, pages 739–753. IEEE, 2019.

[72] M. Fredrikson, S. Jha, and T. Ristenpart. Model inversion attacks that exploit confidence information and basic countermeasures. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, pages 1322–1333, 2015.

[73] N. Carlini, F. Tramer, E. Wallace, M. Jagielski, A. Herbert-Voss, K. Lee, A. Roberts, T. Brown, D. Song, U. Erlingsson, A. Oprea, and C. Raffel. Extracting training data from large language models. In *30th USENIX Security Symposium*, pages 2633–2650, 2021.

[74] A. Krizhevsky and G. Hinton. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009.

[75] Y. Le and X. Yang. Tiny ImageNet visual recognition challenge. Technical report, CS 231N, Stanford University, 2015.

[76] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255. IEEE, 2009.

واژه‌نامه

آ

آداپتر ..... Adapter

آداپتر گلوگاهی ..... Bottleneck Adapter

آداپتر مشترک ..... Shared Adapter

آداپتر مخصوص وظیفه .... Task-Specific Adapter

آداپتر باقی‌مانده ..... Residual Adapter

آنتروپی متقاطع ..... Cross-Entropy

ا

اتصال باقی‌مانده ..... Residual Connection

انطباق کم‌رتبه ..... Low-Rank Adaptation

انرژی تداخل بحرانی Critical Interference Energy

انرژی تعارض گرادیانی . Gradient-Conflict Energy

امتیاز یادگرفتنی ..... Learnable Score

اهمیت پارامتر ..... Parameter Importance

ب

بازپخش تجربه ..... Experience Replay

بازپخش تجربه‌ی تاریک . Dark Experience Replay

بازپخش مولد ..... Generative Replay

بازنشانی ..... Reset

بازه‌ی بوت‌استرپ ..... Bootstrap Interval

بدنه‌ی اصلی ..... Backbone

بیشترین افت ..... WorstDrop

بزرگی گرادیان ..... Gradient Magnitude

پ

پاک‌سازی مبتنی بر فیشر .. Fisher-based Scrubbing

پراکندگی ..... Sparsity

پشتیبان مشترک ..... Shared Support

ت

تابع زیان ..... Loss Function

تابع نفوذ ..... Influence Function

تقطیر دانش ..... Knowledge Distillation

تنظیم پیشوند ..... Prefix-Tuning

تنظیم پرامپت ..... Prompt-Tuning

تنظیم ..... Regularization

تصویر گرادیان ..... Gradient Projection

ترنسفورمر بینایی ..... Vision Transformer

توزیع یکنواخت ..... Uniform Distribution

ح

حالت مقیم ..... Resident State

حافظه‌ی بازپخش ..... Replay Buffer

حافظه‌ی اپیزودیک ..... Episodic Memory

حافظه‌ی اپیزودیک گرادیانی .... Gradient Episodic Memory

حق فراموش‌شدن ..... Right to be Forgotten

حذف داده‌ی گواهی‌شده . Certified Data Removal

حمله‌ی استنتاج عضویت .. Membership Inference Attack

حمله‌ی وارون‌سازی مدل . Model Inversion Attack

حریم خصوصی ..... Privacy

حریم خصوصی تفاضلی ..... Differential Privacy

د

درخواست فراموشی ..... Forgetting Request

دنباله‌ی درخواست ..... Request Schedule

ر

رتبه‌بندی ..... Ranking

مخاطره ..... Risk

مخاطره‌ی تجربی ..... Empirical Risk

ز

زیرشبکه‌ی پراکنده ..... Sparse Subnetwork

زیرشبکه‌ی برنده ..... Winning Subnetwork

س

سیناپس‌های آگاه از حافظه ..... Memory Aware Synapses

سفت‌محافظت‌شده ..... Hard-Protected

ش

شبکه‌ی باقی‌مانده ..... Residual Network

شبکه‌ی پیش‌رونده ..... Progressive Network

ص

صعود گرادیانی ..... Gradient Ascent

صعود گرادیانی بی‌قید ... Unconstrained Gradient Ascent

ض

ضرب‌گر ماسک نرم ..... Soft-Scaling Multiplier

ف

فراموشی انتخابی ..... Selective Forgetting

فراموشی فاجعه‌بار ..... Catastrophic Forgetting

فراموشی تقریبی ..... Approximate Unlearning

فراموشی دقیق ..... Exact Unlearning

فراموشی مرزی ..... Boundary Unlearning

فرضیه‌ی بلیت بخت‌آزمایی ..... Lottery Ticket Hypothesis

گ

گرادیان کاهشی تصادفی ..... Stochastic Gradient Descent

ل

لنگر ..... Anchor

لاجیت ..... Logits

م

ماسک ترکیبی ..... Combined Mask

ماسک دودویی ..... Binary Mask

ماسک نرم ..... Soft Mask

ماسک توجه سخت . Hard Attention to the Task

ماتریس اطلاعات فیشر Fisher Information Matrix

مرجع بازآموزی ..... Retraining Reference

مدل سایه ..... Shadow Model

مقدارگذاری کایمینگ ..... Kaiming Initialization

مزیت عضویت ..... Membership Advantage

میراکردن انتخابی سیناپسی .... Selective Synaptic Dampening

منجمد ..... Frozen

ن

ناحیه‌ی اشتراک ..... Shared Region

نسبت محافظت ..... Protection Ratio

نسبت پارامتر به‌روزشده Updated Parameter Ratio

نرمال‌سازی دسته‌ای ..... Batch Normalization

ه

هم‌پوشانی پارامتری ..... Parameter Overlap

هم‌پوشانی بحرانی ..... Critical Overlap

هم‌پوشانی ساختاری ..... Structural Overlap

هوشمندی سیناپسی ..... Synaptic Intelligence

و

وظیفه‌ی نگه‌داری‌شده ..... Retained Task

ی

یادگیری پیوسته ..... Continual Learning

یادگیری مادام‌العمر ..... Lifelong Learning

یادگیری مادام‌العمر حریم‌خصوصی‌آگاه Privacy-Aware Lifelong Learning

یادگیری کارآمد پارامتری ..... Parameter-Efficient Fine-Tuning

یادگیری افزایشی وظیفه ..... Task-Incremental Learning

یادگیری افزایشی دامنه ..... Domain-Incremental Learning

یادگیری افزایشی کلاس ..... Class-Incremental Learning

یادگیری‌زدایی ماشینی ..... Machine Unlearning

یادگیری‌زدایی برجستگی ..... Saliency Unlearning

پیوست

کد روش EPALL و همه‌ی اسکریپت‌های بازتولید آزمایش‌ها در آدرس زیر قابل دسترس است:

https://github.com/Fatemerjn/EPALL

روش اصلی در متن این پایان‌نامه EPALL نامیده می‌شود و در کد و برچسب‌های آزمایش با شناسه‌ی pall_modified ثبت شده است؛ این دو نام به یک الگوریتم اشاره می‌کنند. پیکربندی کامل آزمایش‌ها در جدول ۵-۱ آمده است.

جدول ۵-۱: پیکربندی کامل آزمایش‌ها. مقادیر میان روش‌ها مشترک‌اند مگر آنکه سطری مخصوص یک روش آن‌ها را بازنویسی کند.

گروه	تنظیم	مقدار
بهینه‌سازی	بهینه‌ساز	SGD، تکانه‌ی 0.9
	نرخ یادگیری	10^{-2} (LoRA روی Split-CIFAR-100: 10^{-3})
	تنظیم وزن	5×10^{-4}
	اندازه‌ی دسته	32
	ایپاک هر وظیفه	20 (استاندارد)، 3 (هم‌پوشانی‌سنگین و پیش‌آموزش‌دیده)
محک‌ها	CIFAR-10	5 وظیفه × 2 کلاس، سطح تصادفی 0.5
	CIFAR-100 استاندارد	10 وظیفه × 10 کلاس مجزای تصادفی، سطح تصادفی 0.1
	CIFAR-100 هم‌پوشانی	10 وظیفه × 5 کلاس، سطح تصادفی 0.2

ادامه در صفحه‌ی بعد

ادامه‌ی جدول ۵-۱

گروه	تنظیم	مقدار
	TinyImageNet	20 وظیفه × 10 کلاس، سطح تصادفی 0.1
معماری	CIFAR-10 /	نسخه‌های ResNet-18
	TinyImageNet / پیش‌آموزش‌دیده	
	CIFAR-100 اصلی	نسخه‌های ResNet-34
	آزمون میان‌معماری	ViT-T/8
بازپخش	بودجه‌ی حافظه	500 نمونه
	تخصیص	از پیش افرازشده؛ *بدون* بازتوزیع پس از حذف
زیرشبکه	پراکندگی Tiny / C10	0.8
	پراکندگی C100	0.9
EPALL	نسبت محافظت ρ	0.2
	ضریب لنگر λ	1.0
	معیار اهمیت	بزرگی گرادیان (پیش‌فرض)؛ انرژی تعارض (مطالعه)
	هدف لنگر	`old` (پیش‌فرض و همه‌ی اجراهای گزارش‌شده)؛ `reinit` فقط در مطالعه‌ی بخش ۴-۱۲-۱
PALL-Adapter	گلوگاه وظیفه / مشترک	16 / 16
	نسبت فراموشی مشترک α_f	0.3
	نسبت محافظت مشترک α_p	0.2
	شدت ماسک نرم p	$\lvert S_{\text{crit}}\rvert/\lvert S_{\text{forget}}\rvert$
	گام فراموشی / ترمیم	50 / 10
LoRA	رتبه / α	16 / 8
مبناها	EWC	λ: 100
	LwF	α: 1.0؛ دمای تقطیر T: 2.0
	DER++	α (logit): 0.5؛ β (CE): 1.0

ادامه در صفحه‌ی بعد

ادامه‌ی جدول ۵-۱

گروه	تنظیم	مقدار
	ER	فقط بازپخش؛ بدون جمله‌ی اضافی
	SSD	آستانه‌ی نسبت فیشر 1.0، شدت میرایی 1.0
	SalUn	نسبت ماسک 0.1، هدف یکنواخت، 50 گام فراموشی
	درخواست فراموشی هر اجرا	3
دنباله‌ها	ساختار	هر وظیفه حداکثر یک‌بار آموزش؛ بدون معرفی دوباره
	الگوی درخواست	CIFAR-10: T T T T F T F F؛ CIFAR-100: T×7 F T T F T F

محیط اجرا. همه‌ی اجراهای گزارش‌شده روی یک NVIDIA RTX 3070 Ti با ۸ گیگابایت حافظه‌ی گرافیکی، دو پردازنده‌ی Xeon E5-2620 v4، ۳۱ گیگابایت حافظه‌ی اصلی، Ubuntu 24.04.4، Python 3.12.3 و PyTorch 2.5.1 انجام شده‌اند. برای بازتولید یک مجموعه نتیجه، گروه متناظر از اسکریپت `tools/run_server_experiments.sh` اجرا می‌شود؛ برای نمونه `bash tools/run_server_experiments.sh g6_standard`. هر گروه پیش از اجرا با سوییچ `--dry-run` فرمان‌های دقیق خود را بدون اجرا چاپ می‌کند. هر مجموعه نتیجه زیر یک برچسب آزمایش مستقل ثبت می‌شود و نگاشت گروه‌ها، برچسب‌ها و تنظیمات در `README` مخزن آمده است. برچسب محلی `thesis-final` نسخه‌ی کنونی پایان‌نامه را نشان نمی‌دهد؛ بنابراین تا زمان تثبیت نسخه‌ی تحویلی، هیچ ادعای بازتولیدپذیری به آن برچسب وابسته نیست.

دنباله‌های درخواست. دنباله‌ها از پیش تولید و در پوشه‌ی `schedules/` ذخیره شده‌اند تا همه‌ی روش‌ها در هر بذر دقیقا یک دنباله را ببینند. الگو به بذر وابسته است: در بذر ۰، دنباله‌های CIFAR-10 و CIFAR-100 به‌ترتیب T T T T F T F F و T×7 F T T F T F هستند؛ در مقابل، دنباله‌ی CIFAR-100 بذر ۵ به‌صورت T×10 F F F است و پس از نخستین حذف آموزشی ندارد. درهم‌سازی SHA-256 فایل‌های بذر ۰ (شانزده نویسه‌ی نخست) چنین است:

`904b860aa0d75436 :cifar10_t5_f3_fixed_seed0.json`

`17ae55e7bf4d629f :cifar100_t10_f3_seed0.json`

Abstract

Continual learning systems may need to unlearn specific information while retaining other tasks. In task-incremental selective forgetting, *parameter overlap* makes this difficult: resetting or repairing weights for one task can damage the others. The Privacy-Aware Lifelong Learning (PALL) framework modularizes task knowledge with stored subnetworks, but at request time it does not distinguish shared coordinates by retained-task sensitivity.

This thesis introduces EPALL, an enhanced PALL procedure that identifies the shared support between the target subnetwork and the union of the active subnetworks, ranks those coordinates by retained-task gradient magnitude, and anchors the high-sensitivity subset with an ℓ_2 penalty during the repair phase. In addition, an exploratory parameter-efficient pathway, PALL-Adapter, is designed and audited through matched component controls.

On Split-CIFAR benchmarks, an eleven-method eight-seed comparison shows that EPALL drives forgotten-task accuracy to the chance level (0.4863 on CIFAR-10 and 0.0959 on CIFAR-100) while matching the full-isolation CLPU reference in retained accuracy to within seed spread. Against the original PALL algorithm, the paired accuracy difference is positive and excludes zero on both datasets (exact one-sided Wilcoxon raw $p = 0.0078$; Holm-adjusted $p = 0.0469$), and survives Holm correction across the six simultaneous tests; WorstDrop on CIFAR-100 falls from 0.0222 to 0.0056, but its CIFAR-10 counterpart does not survive the correction. A 300-run request-position benchmark over five pre-specified grades — a proxy for expected overlap rather than a direct control of it — shows that PALL's WorstDrop grows with the grade while EPALL stays comparatively flat. A complementary sparsity sweep, which moves measured structural overlap by more than tenfold at a fixed target task, finds a positive and significant rank correlation between that overlap and the benefit of protection, though the same knob also changes capacity. Resident-state accounting shows CLPU using 1.6–2.2× as much accounted state.

Matched component controls narrow the mechanism claim, and these limits are reported explicitly: what the evidence supports is retention-sensitivity ranking under a fixed budget, whereas the explicit mask-intersection filter is redundant and the anchor term shows no measurable independent contribution. For PALL-Adapter, the controls attribute task suppression primarily to target-path reset rather than to the overlap-aware soft mask. No result establishes exact, certified, or storage-level removal, and all claims are empirical.

Keywords: Continual Learning, Machine Unlearning, Selective Forgetting, Parameter Overlap, Protected Subnetworks, Parameter-Efficient Fine-Tuning

Sharif University of Technology

Department of Computer Engineering

M.Sc. Thesis

Forgetting Management in Continual Learning Using Deep Learning Methods

By:

Fatemeh Raeejian

Supervisor:

Prof. Hamid Beigy

September 2026