GRASSMANNIAN SUBSPACE ALIGNMENT FOR KNOWLEDGE DISTILLATION

Anonymous authors
Paper under double-blind review

ABSTRACT

In knowledge distillation for neural networks, a compact student learns from a larger teacher while remaining efficient to deploy. When their hidden widths differ, intermediate representations cannot be compared directly, and most feature-based methods learn a projection and then match features coordinate by coordinate. This ties the student to one particular basis of the teacher's representation, even though any rotation of that basis spans the same subspace. We instead align the subspaces themselves: learned projections map student representations into the teacher's space, where we apply point-wise spherical or layer-wise Grassmann subspace alignment. For speech recognition, distilling Whisper Large-v3 into Whisper Medium with the spherical variant and low-rank adaptation (LoRA) reaches 13.49% word error rate (WER) on a LibriSpeech test-other subset while training only 4.55% of the teacher's parameter count. For language modeling, distilling Apertus-8B into a 0.5B student with layer-wise Grassmann alignment shows gains on HellaSwag and PIQA. We further prove that the Grassmann loss has an intuitive interpretation as the expected leakage of student directions outside the teacher subspace.

1 INTRODUCTION

Training a model and deploying it impose different computational requirements. Model compression transfers a complex predictor's behavior to a smaller network (Buciluǎ et al., 2006). Knowledge distillation extends this idea using softened output probabilities, which convey relative preferences among alternatives beyond the correct label (Hinton et al., 2015). This distinction is useful for speech and language models, where a compact student must learn from a substantially larger teacher.

Output supervision accommodates different hidden widths, but does not specify which intermediate structure the student should preserve. Feature hints, attention transfer and relational objectives address this question through different representation targets (Romero et al., 2015; Zagoruyko & Komodakis, 2017; Park et al., 2019). We focus on a simple distinction: a collection of features defines both individual vectors and a subspace. Rotating an orthonormal basis changes those vectors without changing their span. When the target is the span, matching its orthogonal projector avoids choosing a particular basis.

We introduce Grassmannian Alignment, combining output distillation with projection-based representation supervision. A learned lift reconciles student and teacher hidden widths; a Grassmann projector loss compares the resulting subspaces. The lift also defines a pullback metric, connecting the formulation to information geometry and optimization on matrix manifolds (Ay et al., 2017; Absil et al., 2008). For the implemented linear lift, this metric is constant when the parameters are fixed; it is an interpretation of the projection, not an additional loss.

The speech instantiation combines spherical alignment with LoRA: it records 13.49% WER and 7.00% CER on 200 test-other utterances while updating 70.52M parameters, 90.77% fewer than full adaptation of its original student backbone. The language model extension evaluates Grassmann alignment and yields mixed English and multilingual results. Our contribution connects width reconciliation, invariant representation targets and parameter-efficient speech adaptation; the available evaluations do not isolate a recognition gain from geometry alone.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

2 Related Work

2.1 Knowledge Distillation

Model compression (Buciluǎ et al., 2006) and soft-target distillation (Hinton et al., 2015) transfer a teacher's predictive behavior to a student. The original distillation analysis relates high-temperature soft-target matching to squared differences between centered logits; this approximation does not make KL divergence generally Euclidean. Subsequent work examines training duration and consistent teacher–student inputs (Beyer et al., 2022), logit standardization (Sun et al., 2024), and adaptive intermediate distributions that address the teacher–student capacity gap (Shing et al., 2025). Recent byte-model distillation further studies transfer across tokenization schemes (Marathe et al., 2026).

Intermediate supervision includes feature regression (Romero et al., 2015), attention transfer (Zagoruyko & Komodakis, 2017), relational and similarity preservation (Park et al., 2019; Tung & Mori, 2019), and contrastive distillation (Tian et al., 2020). HieRD preserves hierarchical structure in vision–language embeddings (Le et al., 2026). Orthogonal projections are also used in constrained feature distillation by VkD (Miles et al., 2024). SDMD instead learns separate orthonormal teacher and student projections into a shared space compared under an indefinite inner product, which permits sign-flipped alignment (Hayder et al., 2026); the projector loss below is invariant to such sign flips and basis rotations by construction. Our formulation therefore does not claim that projection-based or structure-preserving distillation is new. It uses pooled subspace projectors as the alignment target at selected layers, retaining invariance to the choice of orthonormal basis.

Whisper provides multilingual speech models (Radford et al., 2023), while Distil-Whisper combines pseudo-labeling with architectural compression (Gandhi et al., 2023). LoRA reduces the parameters updated during adaptation (Hu et al., 2022); it does not remove the frozen backbone from inference. On-device systems such as Moonshine and optimized streaming ASR address deployment constraints directly (Jeffries et al., 2024; Banfic et al., 2026). Our experiments report adaptation counts and recognition scores without inferring a deployment improvement from the trainable fraction alone.

2.2 Geometry-Aware Representation Learning

Information geometry describes local divergence structure through the Fisher metric (Amari & Nagaoka, 2000; Amari, 2016; Ay et al., 2017). Riemannian optimization develops updates under manifold constraints (Smith, 1994; Absil et al., 2008). These perspectives distinguish the geometry of output probabilities, hidden representations and model parameters; constraining one does not automatically constrain the others.

A point on $\mathrm{Gr}(k, d)$ is a k-dimensional subspace of $\mathbb{R}^d$, represented by any orthonormal basis U with projector $UU^\top$. A unit vector on a sphere instead retains its direction: changing its sign changes the point but not its one-dimensional span. Decoupled Networks separate angular and magnitude components (Liu et al., 2018b), nGPT normalizes transformer representations on the hypersphere (Loshchilov et al., 2025), and Poincaré embeddings use hyperbolic geometry for hierarchies (Nickel & Kiela, 2017). These objectives preserve different structures. Likewise, Hyper-Connections and manifold-constrained Hyper-Connections modify residual mixing (Zhu et al., 2025; Xie et al., 2025), rather than aligning teacher and student subspaces.

2.3 Grassmann Networks and Positioning

Grassmann discriminant analysis (Hamm & Lee, 2008) and projection metric learning (Huang et al., 2015) established subspace-based learning before deep distillation. Subspaces spanned by a few deep embeddings also serve as class representations in few-shot learning (Simon et al., 2020; Devos & Grossglauser, 2020). GrNet (Huang et al., 2018) combines full-rank mapping, re-orthonormalization and projection mapping in a deep architecture. Our method shares this geometric machinery, but applies a rectangular lift and projector discrepancy to teacher–student hidden states. The current experiments do not compare against an adapted GrNet or VkD baseline, and therefore do not establish superiority to either. Appendix C.1 details the distinction between projection discrepancy, geodesic distance and pullback geometry.

3 METHODOLOGY

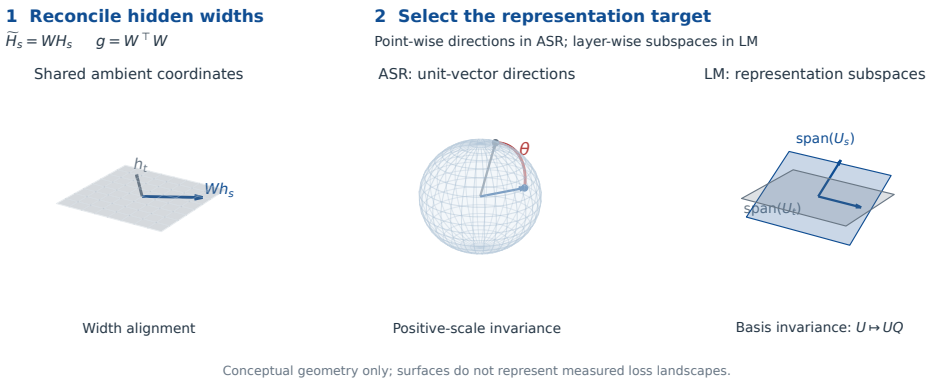

Figure 1: A shared space, two alignment targets. A projection reconciles hidden widths; the loss compares directions in ASR or subspaces in the language model extension. The three surfaces are conceptual illustrations, not measured loss landscapes.

Table 1: Notation for one matched teacher–student layer pair. Batch indices are omitted.

Symbol	Meaning
$H_s \in \mathbb{R}^{d_s \times T}$, $H_t \in \mathbb{R}^{d_t \times T}$	Hidden states; T sequence positions
$W \in \mathbb{R}^{d_t \times d_s}$	Trainable lift into the teacher feature space
$U_s, U_t \in \mathbb{R}^{d_t \times k}$	Orthonormal bases of rank-k subspaces
$P_s = U_s U_s^\top$, $P_t = U_t U_t^\top$	Basis-invariant orthogonal projectors
ℓ_s, ℓ_t; τ	Output logits; distillation temperature

First, map student representations into the teacher's feature space. Next, compare normalized directions in the ASR instantiation or pooled representation subspaces in the language model extension. Finally, combine this signal with output distillation and task supervision; Algorithm 1 in the appendix specifies the implemented language model update.

3.1 PROJECTION

Let $H_s \in \mathbb{R}^{d_s \times T}$ and $H_t \in \mathbb{R}^{d_t \times T}$ contain the student and teacher hidden states as columns. At a matched layer pair, their sequence positions are aligned and $d_s \leq d_t$. Batch indices are omitted. Figure 1 illustrates the geometric motivation of the projection framework.

To resolve the dimensional mismatch, we introduce a learnable linear projection $\widetilde{H}_s = W H_s$, with $W \in \mathbb{R}^{d_t \times d_s}$. The same matrix acts point-wise on all positions at that layer. Different selected layers have separate matrices. The teacher features already lie in the target ambient space and are not transformed by a learned teacher projection.

In the LM implementation, we first average contiguous temporal bins to form k representative columns: $A_s = \mathrm{pool}_k(\widetilde{H}_s)$ and $A_t = \mathrm{pool}_k(H_t)$, both in $\mathbb{R}^{d_t \times k}$. Reduced QR decomposition then gives

$$U_s = \mathrm{qf}(A_s), \qquad U_t = \mathrm{qf}(A_t), \qquad U_s^\top U_s = U_t^\top U_t = I_k. \tag{1}$$

Here qf returns the Q factor, not a unique canonical basis. This interpretation assumes full column rank and $k \leq \min(T, d_s, d_t)$. At rank deficiency, a fixed-size QR factor can include arbitrary completion directions; differentiability and the intended subspace interpretation then require additional care. The current code uses a fixed k, without adaptive rank selection.

Writing $P_s = U_s U_s^\top$ and $P_t = U_t U_t^\top$, the unnormalized projector discrepancy is

$$L_{\text{proj}} = \|P_s - P_t\|_F^2 = 2k - 2\left\|U_s^\top U_t\right\|_F^2 = 2\sum_{j=1}^{k} \sin^2 \theta_j, \tag{2}$$

where θ_j are the principal angles. The implemented loss is $L_{\text{G}} = L_{\text{proj}}/(2k)$, averaged over examples and selected layer pairs. It can be computed using the small matrix $U_s^\top U_t$, without materializing dense $d_t \times d_t$ projectors. It is invariant to $U_s \mapsto U_s Q_s$ and $U_t \mapsto U_t Q_t$ for orthogonal Q_s, Q_t. Equivalently, L_{G} is the expected squared norm of a random unit student direction that falls outside the teacher subspace (Appendix C, Eq. (8)).

The square root of Eq. (2) is a distance between projectors. Its square is a smooth loss, not itself a metric satisfying the triangle inequality. It is also distinct from the intrinsic Grassmann geodesic distance $d_{\text{geo}} = (\sum_j \theta_j^2)^{1/2}$. The two squared quantities agree up to a factor of two locally as the principal angles approach zero. Basis invariance is exact; improved downstream accuracy is an empirical question.

3.2 The Choice of the Manifold

The alignment space depends on the representation being compared. The ASR implementation normalizes projected student and teacher vectors and computes

$$L_{\text{sph}} = \frac{1}{BT}\sum_{b,u} \arccos\big(\text{clip}(\widehat{z}_{s,bu}^\top \widehat{z}_{t,bu}, -1+\epsilon, 1-\epsilon)\big), \quad \widehat{z} = z/\|z\|_2, \tag{3}$$

for nonzero vectors, with $\epsilon = 10^{-7}$ in the main ASR cell. This is a numerically clipped angular distance on the unit sphere. Unlike coordinate MSE, it is invariant to positive rescaling of either nonzero feature vector: supervision targets direction rather than magnitude. This is a structural benefit, not a measured WER gain. It compares corresponding directions, not subspaces; the LM extension uses Eq. (2).

For a smooth map ϕ into a Euclidean ambient space, the pullback tensor is $g_\phi = J_\phi^\top J_\phi$. In the linear case, $g_\phi = W^\top W$, so $\|W(h_1 - h_2)\|_2^2 = (h_1 - h_2)^\top g_\phi (h_1 - h_2)$. This is positive definite if W has full column rank and otherwise only positive semidefinite. A constant positive-definite linear pullback is flat; it does not create a curved space. Furthermore, normalization onto a sphere and quotienting by basis rotations introduce geometry beyond this linear metric. We use this identity to interpret the lift, without equating it to the spherical or Grassmann loss or claiming a separate pullback penalty.

3.3 Loss Design

We combine task supervision, output distillation and representation alignment:

$$\mathcal{L} = w_{\text{CE}}\mathcal{L}_{\text{CE}} + w_{\text{KL}}\mathcal{L}_{\text{KL}} + w_{\text{geo}}\mathcal{L}_{\text{geo}}. \tag{4}$$

For language modeling, $\mathcal{L}_{\text{CE}}$ is next-token cross-entropy and

$$\mathcal{L}_{\text{KL}} = \tau^2 \,\text{mean}_{b,u}\,\text{KL}(\text{softmax}(\ell_{t,bu}/\tau) \parallel \text{softmax}(\ell_{s,bu}/\tau)), \qquad \tau = 2. \tag{5}$$

The teacher is evaluated without gradients. Geometry acts on hidden representations, whereas KL acts on output distributions. Figure 2 illustrates these two complementary supervision branches.

In ASR, CE supplies transcript supervision, KL transfers output preferences, and angular alignment transfers feature directions independently of positive feature scaling. Two additional terms match temporal second-difference magnitudes and contrast corresponding frames against other frames. These terms encourage complementary structure; their separate recognition benefits are not measured. They neither impose a Grassmann trajectory nor implement Sinkhorn matching. Appendix A specifies the curriculum and ASR-specific KL reduction.

Optimization is implementation-specific. ASR uses AdamW on LoRA parameters and one unconstrained linear projection. The LM reference code uses Euclidean optimizer updates and reorthonormalizes each W_i by QR every 50 optimizer steps. Thus $W_i^\top W_i = I$ immediately after that operation, up to numerical precision, but need not remain so between operations. This is periodic constraint restoration, not intrinsic Grassmann optimization or a Cayley update at every step.

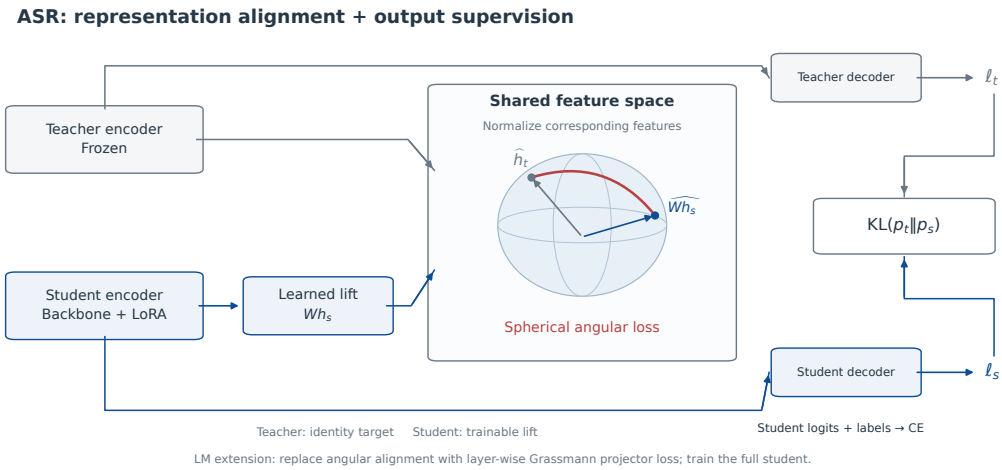

Figure 2: Representation and output supervision are complementary. In ASR, a learned student projection supports spherical angular alignment, while teacher and student decoder logits enter KL distillation. The curved surface is schematic. The language model extension replaces the central point-wise loss with layer-wise Grassmann projector alignment and trains the full student.

4 Experimental Study

4.1 Setup

The speech teacher is Whisper Large-v3 (approximately 1.55B parameters, hidden width 1280); the student is Whisper Medium (width 1024). Both original backbones are frozen. Rank-64 LoRA adapters in the student encoder and decoder, together with a $1024 \rightarrow 1280$ projection, contribute 70,516,736 trainable parameters, approximately 4.55% of the teacher's nominal size. The student contains 833,063,936 parameters including adapters, 46.25% fewer than the nominal teacher (a $1.86\times$ parameter reduction). The teacher and projection are training-only; LoRA reduces updated parameters, not the frozen backbone retained at inference.

The ASR code interleaves seven Multilingual LibriSpeech language streams (Pratap et al., 2020), using AdamW, batch size one and at most 20,000 updates with an NVIDIA H100 80GB HBM3 GPU identified in the saved main-cell session. The evaluated checkpoint is from step 17,600. Evaluation processes 200 LibriSpeech test-other utterances successfully (Panayotov et al., 2015), with lowercasing, ASCII-punctuation removal and whitespace normalization. Its GPU and selected utterance IDs are not recorded. Appendix E gives the sampling rules and parameter breakdown; Appendix A gives the curriculum.

The language model teacher is Apertus-8B-2509 (Swiss AI Initiative, 2025). The reference script trains the full Apertus-v1.1-0.5B student, with four matched layer pairs and rank $k = 32$. It uses a 4-bit NF4 teacher, bfloat16 student and float32 QR. Training defaults specify FineWeb-Edu `sample-10BT` (Penedo et al., 2024) and a nominal 6M-token budget; complete manifests linking these defaults to the evaluated checkpoints are unavailable.

Both LM evaluations record 438,349,352 student parameters, an NVIDIA GH200 120GB GPU, zero-shot prompts and a 4,096-token context in LM Evaluation Harness v0.4.13 (EleutherAI, 2026). ARC-Easy and ARC-Challenge (Clark et al., 2018), HellaSwag (Zellers et al., 2019), WinoGrande (Sakaguchi et al., 2020) and PIQA (Bisk et al., 2020) each use 1,000 examples. XNLI (Conneau et al., 2018) uses 1,000 examples per language; XCOPA (Ponti et al., 2020) uses 500. Appendix E records splits, language codes and evaluator settings.

4.2 Experiments

Table 2 reports the available test-other WER comparison. Ours updates 70.52M parameters rather than the original student's 763.86M, a $10.83\times$ reduction in updated parameter count. This saving

comes from LoRA and freezing; geometry supplies the representation target. Its 13.49% WER exceeds most published reference values, but different test subsets prevent ranking the systems. No matched CE/KL-only or MSE evaluation isolates geometry's accuracy contribution.

Table 2: LibriSpeech test-other WER. Published full-split values are from Gandhi et al. (2023, Table 16). Shading identifies Ours, evaluated on 200 utterances with different normalization. Published rows provide context, not a matched ranking. Ours also records 7.00% CER on the same subset; no test-clean result is available.

Model	WER (%)	Evaluation scope
Whisper Tiny.en	14.1	Full split
Whisper Base.en	10.4	Full split
Whisper Small.en	7.4	Full split
Whisper Medium.en	6.1	Full split
Whisper Large-v2	5.6	Full split
Distil-Medium.en	8.0	Full split
Distil-Large-v2	6.9	Full split
Grassmannian Alignment (Ours, Whisper Medium)	13.49	200 utterances

Tables 3–6 compare our runs and retain every raw task and language score. Differences are in percentage points; reported standard errors describe evaluation sampling, not training-seed variability.

Table 3: Zero-shot LM accuracy (%). Values after $\pm$ are the standard errors reported by the evaluation harness, not variation across training seeds. Δ is Grassmannian Alignment minus Baseline in percentage points. XNLI and XCOPA are equal-size language aggregates.

Task	n	Baseline (ours)	Grassmannian Alignment (ours)	Δ
ARC-Easy	1,000	26.40 ± 1.39	26.90 ± 1.40	+0.50
ARC-Challenge	1,000	22.00 ± 1.31	20.40 ± 1.27	-1.60
HellaSwag	1,000	27.80 ± 1.42	28.70 ± 1.43	+0.90
WinoGrande	1,000	51.00 ± 1.58	48.10 ± 1.58	-2.90
PIQA	1,000	50.20 ± 1.58	52.40 ± 1.58	+2.20
XNLI	15,000	33.43 ± 0.39	33.17 ± 0.38	-0.27
XCOPA	5,500	52.53 ± 0.67	51.80 ± 0.67	-0.73

The raw metrics preserve all seven task groups; Table 4 instead combines ARC-Easy and ARC-Challenge into one equally weighted ARC score for a six-group summary. Neither mean measures statistical significance. Length-normalized scores remain in Table 14; Figure 5 shows their effect on the comparison.

Our six-group means are 39.86% for Baseline and 39.64% for Grassmannian Alignment, respectively 0.97 and 0.75 percentage points above nominal chance (38.89%). The alignment configuration gains 0.90 points on HellaSwag and 2.20 on PIQA, but loses 0.55 on combined ARC, 2.90 on WinoGrande, 0.27 on XNLI and 0.73 on XCOPA. The six-group mean declines by 0.22 points: gains on selected tasks do not support an overall improvement.

The nominal 6M-token configuration is 3.53×10^{-6} of the 1.7T-token Apertus-v1.1 budget (Panferov et al., 2026). At 32,768 tokens per optimizer update, it corresponds to approximately 183 update-equivalents, before rounding to complete updates. Missing run manifests prevent confirming the exact consumed budget for the evaluated checkpoints. The comparison does not evaluate either objective at convergence or isolate training budget as the cause of the observed gap.

Table 4: Our pilot models and published base-model references (accuracy, %). Shading identifies our runs. Reference scores and reported means are copied from Panferov et al. (2026, Table 3); they are contextual comparisons, not matched reruns. For ours, ARC averages raw ARC-Easy/Challenge accuracy and Mean averages six groups. Tokens are nominal for ours; reference protocols and ARC aggregation are not established as identical.

Model	Tokens	ARC	HS	WG	XNLI	XCOPA	PIQA	Mean
Nominal chance	–	25.00	25.00	50.00	33.33	50.00	50.00	38.89
Baseline (ours)	6M	24.20	27.80	51.00	33.43	52.53	50.20	39.86
Grassmannian Alignment (ours)	6M	23.65	28.70	48.10	33.17	51.80	52.40	39.64
Apertus-v1.1-0.5B	1.7T	44.96	40.42	57.06	41.51	55.49	71.27	51.79
Qwen3-0.6B-Base	–	48.35	41.01	59.20	39.55	54.96	70.29	52.23
Apertus-v1.1-1.5B	1.7T	52.66	48.31	61.72	42.94	59.76	74.54	56.66
Qwen3-1.7B-Base	–	56.49	49.36	63.38	41.66	58.35	75.79	57.51
SmolLM2-1.7B	–	60.23	53.38	66.22	37.57	53.51	77.10	58.00

HS denotes HellaSwag and WG denotes WinoGrande. Dashes denote omitted token budgets. Nominal chance assumes four ARC choices; individual ARC items can differ.

Table 5: XNLI by language (%); $n = 1{,}000$ each. B and GA denote our Baseline and Grassmannian Alignment.

Language	B	GA	Δ
ar	33.30	30.60	-2.70
bg	34.20	33.90	-0.30
de	33.50	34.00	+0.50
el	33.10	33.30	+0.20
en	35.30	31.60	-3.70
es	33.30	33.30	+0.00
fr	33.30	33.30	+0.00
hi	33.10	35.60	+2.50
ru	32.10	33.70	+1.60
sw	33.80	32.80	-1.00
th	33.40	33.60	+0.20
tr	33.30	33.30	+0.00
ur	33.30	33.30	+0.00
vi	32.30	33.40	+1.10
zh	34.20	31.80	-2.40

Table 6: XCOPA by language (%); $n = 500$ each. Δ is GA minus B in percentage points.

Language	B	GA	Δ
et	49.40	48.40	-1.00
ht	52.60	51.20	-1.40
id	52.20	52.20	+0.00
it	51.00	49.20	-1.80
qu	50.40	50.20	-0.20
sw	51.40	53.20	+1.80
ta	57.00	56.60	-0.40
th	55.00	54.00	-1.00
tr	53.40	50.80	-2.60
vi	52.80	52.00	-0.80
zh	52.60	52.00	-0.60

Full language-wise standard errors are retained in Appendix I.

4.3 Results and Evaluation

Our XNLI means (33.43/33.17%) lie near one-third chance; XCOPA (52.53/51.80%) lies within three points of 50%. Grassmannian Alignment is lower on nine of eleven XCOPA languages. Near-chance XNLI and the predominantly negative XCOPA differences provide no reliable evidence of improved multilingual transfer.

Published Apertus-v1.1-0.5B scores are 41.51% on XNLI and 55.49% on XCOPA, both above our pilot scores. Different training budgets and incompletely matched protocols prevent attributing these gaps to the alignment objective. Figure 3 retains every within-pilot task comparison.

Language-level changes are heterogeneous (Figure 4): XNLI ranges from -3.7 points in English to $+2.5$ in Hindi; XCOPA ranges from -2.6 in Turkish to $+1.8$ in Swahili. Languages share checkpoints and may contain translated examples, so they are not independent training repetitions.

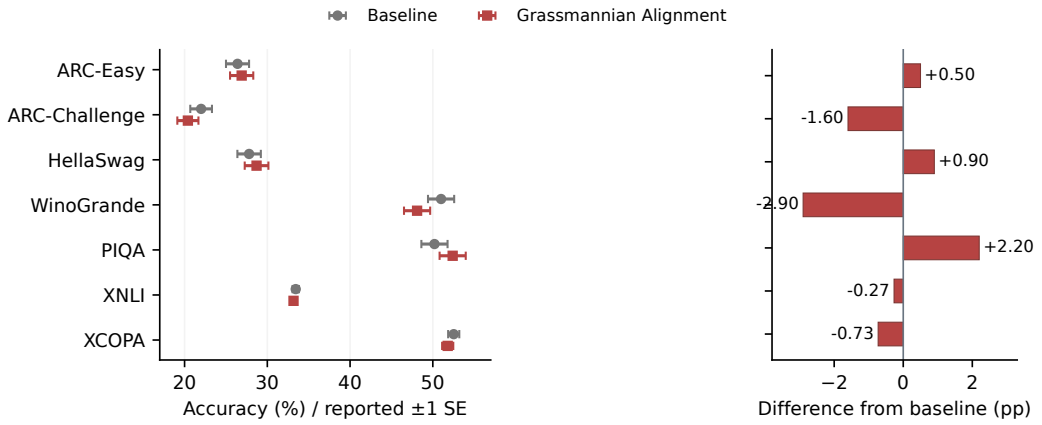

Figure 3: Our two models: LM accuracy and Grassmannian Alignment–Baseline differences from the evaluation exports. Left: accuracy with reported standard errors. Right: signed changes in percentage points; blue denotes gains and muted red denotes losses. The mixture of positive and negative changes is visible across the full task suite.

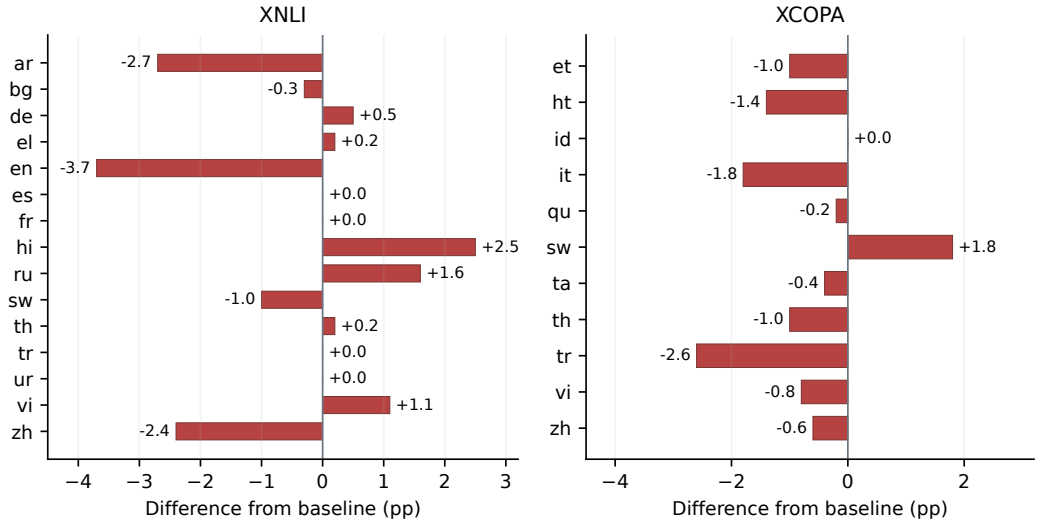

Figure 4: Per-language changes in raw accuracy. Differences are Grassmannian Alignment minus Baseline; blue denotes gains and muted red denotes losses. They are not significance estimates.

The baseline fixes the weights at $0.1L_{\mathrm{CE}} + 0.9L_{\mathrm{KL}}$. Grassmannian Alignment changes both the representation loss and the CE/KL schedule. Consequently, this comparison measures a combined configuration change; isolating geometry requires matched curricula and repeated seeds.

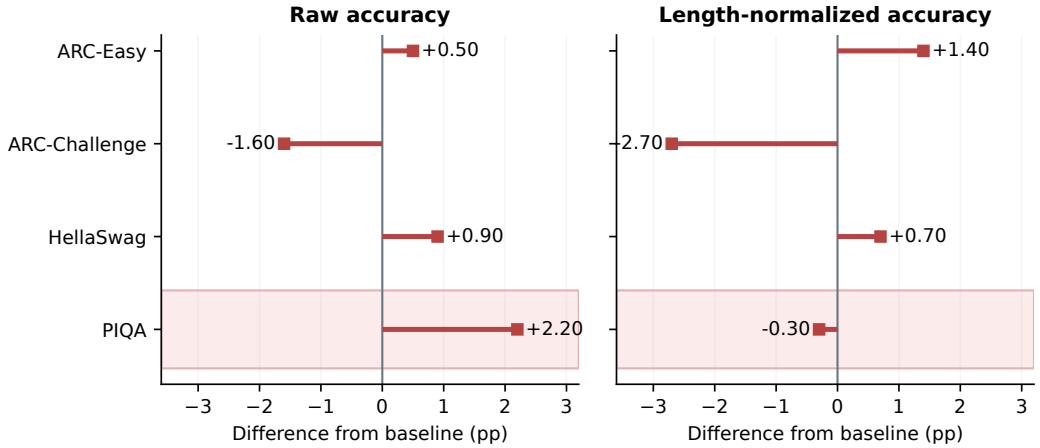

Figure 5: A raw-accuracy gain need not survive length normalization. The same four task evaluations are shown using both metrics. Bars extending right of zero favor Grassmannian Alignment. PIQA changes from $+2.20$ points in raw accuracy to -0.30 points after length normalization; the shaded row makes that reversal explicit. Both panels use the same checkpoints and task examples.

Metric choice further affects the comparison: PIQA improves by 2.20 points in raw accuracy but declines by 0.30 after length normalization. ARC-Easy and HellaSwag improve under both metrics, while ARC-Challenge declines under both. These outcomes support task-specific analysis rather than a uniform transfer claim.

5 Discussion and Limitations

The experiments separate two representation targets: normalized directions in speech and subspace spans in language modeling. Only the latter directly evaluates Grassmann projector alignment. Fixed-rank QR also requires adequate rank and conditioning, and matching subspaces alone does not rule out representation collapse. A controlled follow-up should vary geometry at a fixed curriculum and evaluate the same checkpoints on complete speech test sets and paired language-model examples.

6 Conclusion

We present Grassmannian Alignment for transferring intermediate representations across teacher and student models with different hidden dimensions. Learnable projections reconcile their widths, while spherical and Grassmann objectives preserve different aspects of representation structure. The ASR configuration records 13.49% WER with 90.77% fewer updated parameters than full student adaptation; its student retains 46.25% fewer parameters than the nominal teacher. These counts distinguish adaptation efficiency from model compression. The LM six-group mean is 39.64%, versus 39.86% for our baseline, with no established multilingual advantage. Matched curricula and repeated seeds remain necessary to isolate geometry's accuracy contribution.

AI USE STATEMENT

AI tools assisted with figure preparation, portions of language editing, technical consistency checks, supplementary mathematical exposition, descriptive analyses, plotting code and LaTeX layout. The surface artwork in Figures 1 and 2 came from the authors' existing materials, and the plotted measurements came from the recorded evaluations. No experimental measurements were generated by AI. The authors retain responsibility for the scientific content and for manually reviewing, revising and approving the final manuscript.

REPRODUCIBILITY STATEMENT

The accompanying source package contains the final figure assets and numerical table inputs. Section 4.1 describes the datasets, hardware, and evaluation settings. Appendix A details the implemented losses and training schedules, Appendix B specifies the language model training algorithm, and Appendix C derives the projector identities. The accompanying anonymized code package provides training and evaluation scripts, environment specifications, and configuration files, with documented limitations and differences from historical experiment settings. Complete training manifests, paired language model predictions, and ASR reference and hypothesis transcripts for the reported runs are not available in the retained experimental records.

REFERENCES

P.-A. Absil, Robert Mahony, and Rodolphe Sepulchre. *Optimization Algorithms on Matrix Manifolds*. Princeton University Press, 2008.

Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2013. ACM SIGKDD, ACM. doi: 10.1145/2487575. URL https://doi.org/10.1145/2487575.

Shun-ichi Amari. *Information Geometry and Its Applications*, volume 194. Springer, 2016. doi: 10.1007/978-4-431-55978-8. URL https://link.springer.com/book/10.1007/978-4-431-55978-8.

Shun-ichi Amari and Hiroshi Nagaoka. *Methods of information geometry*, volume 191. American Mathematical Soc., 2000.

Nihat Ay, Jürgen Jost, Hông Vân Lê, and Lorenz Schwachhöfer. *Information Geometry*, volume 64 of *Ergebnisse der Mathematik und ihrer Grenzgebiete. 3. Folge*. Springer, 2017. doi: 10.1007/978-3-319-56478-4. URL https://link.springer.com/book/10.1007/978-3-319-56478-4.

Nenad Banfic, David Fan, Kunal Vaishnavi, Sam Kemp, Sunghoon Choi, Rui Ren, Sayan Shaw, and Meng Tang. Pushing the limits of on-device streaming ASR: A compact, high-accuracy english model for low-latency inference. *arXiv preprint arXiv:2604.14493*, 2026. URL https://arxiv.org/abs/2604.14493.

Lucas Beyer, Xiaohua Zhai, Amélie Royer, Larisa Markeeva, Rohan Anil, and Alexander Kolesnikov. Knowledge distillation: A good teacher is patient and consistent. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10925–10934, 2022. URL https://openaccess.thecvf.com/content/CVPR2022/html/Beyer_Knowledge_Distillation_A_Good_Teacher_Is_Patient_and_Consistent_CVPR_2022_paper.html.

Yonatan Bisk, Rowan Zellers, Ronan Le Bras, Jianfeng Gao, and Yejin Choi. PIQA: Reasoning about physical commonsense in natural language. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2020. URL https://arxiv.org/abs/1911.11641.

Cristian Buciluǎ, Rich Caruana, and Alexandru Niculescu-Mizil. Model compression. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 535–541. ACM, 2006. doi: 10.1145/1150402.1150464. URL https://www.cs.cornell.edu/~caruana/compression.kdd06.pdf.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? try ARC, the AI2 reasoning challenge. *arXiv preprint arXiv:1803.05457*, 2018. URL https://arxiv.org/abs/1803.05457.

Alexis Conneau, Ruty Rinott, Guillaume Lample, Adina Williams, Samuel R. Bowman, Holger Schwenk, and Veselin Stoyanov. XNLI: Evaluating cross-lingual sentence representations. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 2018. URL https://aclanthology.org/D18-1269/.

Arnout Devos and Matthias Grossglauser. Regression networks for meta-learning few-shot classification. In *7th ICML Workshop on Automated Machine Learning*, 2020. URL https://arxiv.org/abs/1905.13613.

EleutherAI. A framework for few-shot language model evaluation, 2026. URL https://github.com/EleutherAI/lm-evaluation-harness. Software, evaluated version 0.4.13.

William Fedus, Barret Zoph, and Noam Shazeer. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *Journal of Machine Learning Research*, 23(120):1–39, 2022.

Sanchit Gandhi, Patrick von Platen, and Alexander M. Rush. Distil-Whisper: Robust knowledge distillation via large-scale pseudo labelling. *arXiv preprint arXiv:2311.00430*, 2023. URL https://arxiv.org/abs/2311.00430.

Jihun Hamm and Daniel D Lee. Grassmann discriminant analysis: a unifying view on subspace-based learning. In *Proceedings of the 25th international conference on Machine learning*, pp. 376–383, 2008. URL https://icml.cc/Conferences/2008/papers/312.pdf.

Felix Hausdorff. *Grundzüge der Mengenlehre*. Veit & Comp, 1914. URL https://www.mathematics.uni-bonn.de/hcm/felix-hausdorff/schriftenverzeichnis.

Zeeshan Hayder, Ali Cheraghian, Lars Petersson, and Mehrtash Harandi. Subspace-guided knowledge distillation for efficient model transfer. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 5057–5066, 2026.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015. URL https://arxiv.org/abs/1503.02531. Presented at the NIPS 2014 Deep Learning Workshop; arXiv release 2015.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022. URL https://arxiv.org/abs/2106.09685.

Zhiwu Huang, Ruiping Wang, Shiguang Shan, and Xilin Chen. Projection metric learning on grassmann manifold with application to video based face recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 140–149, 2015. URL https://openaccess.thecvf.com/content_cvpr_2015/html/Huang_Projection_Metric_Learning_2015_CVPR_paper.html.

Zhiwu Huang, Jiqing Wu, and Luc Van Gool. Building deep networks on Grassmann manifolds. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2018. URL https://arxiv.org/abs/1611.05742.

Nat Jeffries, Evan King, Manjunath Kudlur, Guy Nicholson, James Wang, and Pete Warden. Moonshine: Speech recognition for live transcription and voice commands. *arXiv preprint arXiv:2410.15608*, 2024. URL https://arxiv.org/abs/2410.15608.

Albert Q. Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, Gianna Lengyel, Guillaume Bour, Guillaume Lample, Lélio Renard Lavaud, Lucile Saulnier, Marie-Anne Lachaux, Pierre Stock, Sandeep Subramanian, Sophia Yang, Szymon Antoniak,

Teven Le Scao, Théophile Gervet, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mixtral of experts. *arXiv preprint arXiv:2401.04088*, 2024. URL https://arxiv.org/abs/2401.04088.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020. URL https://arxiv.org/abs/2001.08361.

Aaron Klein and Frank Hutter. Tabular benchmarks for joint architecture and hyperparameter optimization. *arXiv preprint arXiv:1905.04970*, 2019. URL https://arxiv.org/abs/1905.04970.

Amandeep Kumar and Vishal M. Patel. Learning on the manifold: Unlocking standard diffusion transformers with representation encoders. *arXiv preprint arXiv:2602.10099*, 2026. URL https://arxiv.org/abs/2602.10099.

Vinh Le, Nguyen Dang, Tu Vu, Linh Van, Duc Nguyen, and Trung Le. HieRD: Hierarchical relational distillation for vision-language embedding models. In *International Conference on Machine Learning*, 2026. URL https://icml.cc/virtual/2026/poster/63112.

Ruishan Liu, Nicolo Fusi, and Lester Mackey. Teacher-student compression with generative adversarial networks. *arXiv preprint arXiv:1812.02271*, 2018a. URL https://arxiv.org/abs/1812.02271.

Weiyang Liu, Zhen Liu, Zhiding Yu, Bo Dai, Rongmei Lin, Yisen Wang, James M. Rehg, and Le Song. Decoupled networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2771–2779, 2018b. URL https://arxiv.org/abs/1804.08071.

Ilya Loshchilov, Cheng-Ping Hsieh, Simeng Sun, and Boris Ginsburg. nGPT: Normalized transformer with representation learning on the hypersphere. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=se4vjm7h4E.

Pan Lu, Swaroop Mishra, Tony Xia, Liang Qiu, Kai-Wei Chang, Song-Chun Zhu, Oyvind Tafjord, Peter Clark, and Ashwin Kalyan. Learn to explain: Multimodal reasoning via thought chains for science question answering. In *Advances in Neural Information Processing Systems*, 2022. URL https://arxiv.org/abs/2209.09513.

Kalyani Marathe, Artidoro Pagnoni, Tomasz Limisiewicz, Margaret Li, Mike Lewis, Luke Zettlemoyer, and Srinivasan Iyer. Breaking the token ceiling: Distilling smaller, stronger byte models. *arXiv preprint arXiv:2609.12303*, 2026. URL https://arxiv.org/abs/2609.12303.

Ahmed Masry, Do Xuan Long, Jia Qing Tan, Shafiq Joty, and Enamul Hoque. ChartQA: A benchmark for question answering about charts with visual and logical reasoning. In *Findings of the Association for Computational Linguistics: ACL 2022*, 2022. URL https://arxiv.org/abs/2203.10244.

Minesh Mathew, Dimosthenis Karatzas, and C. V. Jawahar. DocVQA: A dataset for VQA on document images. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2021. URL https://arxiv.org/abs/2007.00398.

Roy Miles, Ismail Elezi, and Jiankang Deng. VkD: Improving knowledge distillation using orthogonal projections. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 15720–15730, 2024. URL https://arxiv.org/abs/2403.06213.

Moonshine AI. Moonshine: On-device speech recognition, 2026. URL https://github.com/moonshine-ai/moonshine. Software repository, accessed September 25, 2026.

Siyuan Mu and Sen Lin. A comprehensive survey of mixture-of-experts: Algorithms, theory, and applications. *arXiv preprint arXiv:2503.07137*, 2025. URL https://arxiv.org/abs/2503.07137.

Maximilian Nickel and Douwe Kiela. Poincaré embeddings for learning hierarchical representations. *Advances in neural information processing systems*, 30, 2017.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018. URL https://arxiv.org/abs/1807.03748.

Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur. LibriSpeech: An ASR corpus based on public domain audio books. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 5206–5210, 2015. URL https://www.danielpovey.com/files/2015_icassp_librispeech.pdf.

Andrei Panferov, Davit Melikidze, Martin Jaggi, and Dan Alistarh. Apertus llm family expansion via distillation and quantization. *arXiv preprint arXiv:2605.29128*, 2026. URL https://arxiv.org/abs/2605.29128v2.

Wonpyo Park, Dongju Kim, Yan Lu, and Minsu Cho. Relational knowledge distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3967–3976, 2019. URL https://arxiv.org/abs/1904.05068.

Guilherme Penedo, Hynek Kydlíček, Loubna Ben Allal, Anton Lozhkov, Margaret Mitchell, Colin Raffel, Leandro von Werra, and Thomas Wolf. The FineWeb datasets: Decanting the web for the finest text data at scale. *arXiv preprint arXiv:2406.17557*, 2024. URL https://arxiv.org/abs/2406.17557.

Cristian Perez Jensen and Seyedmorteza Sadat. Efficient distillation of classifier-free guidance using adapters. *Transactions on Machine Learning Research*, 2025. URL https://openreview.net/forum?id=uMz8FiW01.

Edoardo Maria Ponti, Goran Glavaš, Olga Majewska, Qianchu Liu, Ivan Vulić, and Anna Korhonen. XCOPA: A multilingual dataset for causal commonsense reasoning. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, 2020. URL https://aclanthology.org/2020.emnlp-main.185/.

Vineel Pratap, Qiantong Xu, Anuroop Sriram, Gabriel Synnaeve, and Ronan Collobert. MLS: A large-scale multilingual dataset for speech research. In *Interspeech*, pp. 2757–2761, 2020. URL https://www.isca-archive.org/interspeech_2020/pratap20_interspeech.html.

Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. Robust speech recognition via large-scale weak supervision. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202, pp. 28492–28518, 2023. URL https://proceedings.mlr.press/v202/radford23a.html.

Adriana Romero, Nicolas Ballas, Samira Ebrahimi Kahou, Antoine Chassang, Carlo Gatta, and Yoshua Bengio. FitNets: Hints for thin deep nets. In *International Conference on Learning Representations*, 2015. URL https://arxiv.org/abs/1412.6550.

Evgenia Rusak, Patrik Reizinger, Attila Juhos, Oliver Bringmann, Roland S. Zimmermann, and Wieland Brendel. InfoNCE: Identifying the gap between theory and practice. In *Proceedings of the 28th International Conference on Artificial Intelligence and Statistics*, volume 258 of *Proceedings of Machine Learning Research*, pp. 4159–4167, 2025. URL https://proceedings.mlr.press/v258/rusak25a.html.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. WinoGrande: An adversarial winograd schema challenge at scale. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2020. URL https://arxiv.org/abs/1907.10641.

Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. *arXiv preprint arXiv:1701.06538*, 2017. URL https://arxiv.org/abs/1701.06538.

Makoto Shing, Kou Misaki, Han Bao, Sho Yokoi, and Takuya Akiba. TAID: Temporally adaptive interpolated distillation for efficient knowledge transfer in language models. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=cqsw28DuMW.

Christian Simon, Piotr Koniusz, Richard Nock, and Mehrtash Harandi. Adaptive subspaces for few-shot learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4135–4144, 2020. doi: 10.1109/CVPR42600.2020.00419.

Steven T. Smith. Optimization techniques on riemannian manifolds. In *Hamiltonian and Gradient Flows, Algorithms and Control*, volume 3 of *Fields Institute Communications*, pp. 113–135. American Mathematical Society, 1994. URL https://arxiv.org/abs/1407.5965. Author manuscript deposited on arXiv in 2014.

Shangquan Sun, Wenqi Ren, Jingzhi Li, Rui Wang, and Xiaochun Cao. Logit standardization in knowledge distillation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 15731–15740, 2024.

Swiss AI Initiative. Apertus-8B-2509 model card, 2025. URL https://huggingface.co/swiss-ai/Apertus-8B-2509. Public teacher model release.

Keyu Tian, Yi Jiang, Zehuan Yuan, Bingyue Peng, and Liwei Wang. Visual autoregressive modeling: Scalable image generation via next-scale prediction. *Advances in neural information processing systems*, 37:84839–84865, 2024.

Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive representation distillation. In *International Conference on Learning Representations*, 2020. URL https://arxiv.org/abs/1910.10699.

Frederick Tung and Greg Mori. Similarity-preserving knowledge distillation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 1365–1374, 2019.

Vaibhavs10 and contributors. Insanely fast whisper, 2026. URL https://github.com/Vaibhavs10/insanely-fast-whisper. Software repository, accessed September 25, 2026.

Zhenda Xie, Yixuan Wei, Huanqi Cao, Chenggang Zhao, Chengqi Deng, Jiashi Li, Damai Dai, Huazuo Gao, Jiang Chang, Liang Zhao, Shangyan Zhou, Zhean Xu, Zhengyan Zhang, Wangding Zeng, Shengding Hu, Yuqing Wang, Jingyang Yuan, Lean Wang, and Wenfeng Liang. mHC: Manifold-constrained hyper-connections. *arXiv preprint arXiv:2512.24880*, 2025. URL https://arxiv.org/abs/2512.24880. First posted December 2025.

Shuo Yang, Lu Liu, and Min Xu. Free lunch for few-shot learning: Distribution calibration. In *International Conference on Learning Representations*, 2021. URL https://arxiv.org/abs/2101.06395.

Jiazuo Yu, Yunzhi Zhuge, Lu Zhang, Ping Hu, Dong Wang, Huchuan Lu, and You He. Boosting continual learning of vision-language models via mixture-of-experts adapters. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 23219–23230, 2024.

Xiang Yue, Yuansheng Ni, Kai Zhang, Tianyu Zheng, Ruoqi Liu, Ge Zhang, Samuel Stevens, Dongfu Jiang, Weiming Ren, Yuxuan Sun, Cong Wei, Botao Yu, Ruibin Yuan, Renliang Sun, Ming Yin, Boyuan Zheng, Zhenzhu Yang, Yibo Liu, Wenhao Huang, Huan Sun, Yu Su, and Wenhu Chen. MMMU: A massive multi-discipline multimodal understanding and reasoning benchmark for expert AGI. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024. URL https://arxiv.org/abs/2311.16502.

Sergey Zagoruyko and Nikos Komodakis. Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer. In *International Conference on Learning Representations*, 2017. URL https://arxiv.org/abs/1612.03928.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. HellaSwag: Can a machine really finish your sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 2019. URL https://arxiv.org/abs/1905.07830.

Tao Zhong, Zhixiang Chi, Li Gu, Yang Wang, Yuanhao Yu, and Jin Tang. Meta-dmoe: Adapting to domain shift by meta-distillation from mixture-of-experts. *Advances in Neural Information Processing Systems*, 35:22243–22257, 2022.

Defa Zhu, Hongzhi Huang, Zihao Huang, Yutao Zeng, Yunyao Mao, Banggu Wu, Qiyang Min, and Xun Zhou. Hyper-connections. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/f1e8ff97057e0c3f6bcb7018c78d4df1-Abstract-Conference.html.

A Implementation Details

The main ASR training cell uses the final encoder hidden states, one bias-free linear projection, and Eq. (3). With $\widehat{z}_u$ denoting a normalized feature at time u, the temporal regularizer is

$$\kappa_u(z) = \|\widehat{z}_{u+2} - 2\widehat{z}_{u+1} + \widehat{z}_u\|_2\,, \qquad L_{\text{temp}} = \text{mean}_{b,u}\left(\kappa_u(z_s) - \kappa_u(z_t)\right)^2. \tag{6}$$

It compares finite-difference magnitudes along time; it is not an intrinsic curvature estimator with a metric connection. The contrastive loss averages student-to-teacher and teacher-to-student cross-entropies over corresponding normalized frame vectors, with temperature 0.07 and at most 2,048 selected positions. Table 7 records the main cell's weights. The contrastive weight decays linearly during stage II.

Table 7: ASR main-cell curriculum. All stages use $w_{\text{CE}} = 1$. Step intervals describe the configured schedule.

Stage (steps)	Learning rate	w_{KL}	w_{sph}	w_{temp}	w_{cont}
I: $[0, 2000)$	7×10^{-5}	0.50	0.15	0.20	0.30
II: $[2000, 16000)$	2×10^{-5}	0.05	0.05	0.10	$0.30 \to 0$
III: $[16000, 20000)$	10^{-5}	0.01	0.01	0	0

The code uses temperature four and PyTorch's `batchmean` reduction, without the τ^2 prefactor or averaging over sequence positions. Consequently, its numerical weights cannot be transferred directly to the token-mean LM loss. The teacher receives `decoder_input_ids=labels`, while the student receives `labels=labels`; decoder shifting and vocabulary alignment must be checked before treating these logits as identically conditioned targets in a rerun. The training code truncates the vocabulary axis to the smaller size rather than recording a tokenizer mapping. These are implementation limitations, not additional proposed method components.

For $T \geq k$, the code divides sequence positions into k contiguous intervals with boundaries $\lfloor jT/k \rfloor$, averages features in each interval and applies reduced QR. For $T < k$, it repeats the last position to length k; this padding cannot create rank and does not resolve the rank-deficiency caveat. The geometry loss is averaged over four selected layer pairs. The student is trainable in full; LoRA is specific to the ASR experiment. Each lift is initialized by QR and restored by QR every 50 optimizer steps. Between restorations its parameters follow Euclidean optimizer updates. There is no extra pullback-metric loss or hyperbolic distance term.

For consumed-token fraction p, the reference script uses

$$(w_{\text{CE}}, w_{\text{KL}}, w_{\text{G}}) = \begin{cases} (0, 0.5, 0.5), & 0 \leq p < 0.2, \\ (1, 0.1, 0.3[1 - (p - 0.2)/0.4]), & 0.2 \leq p < 0.6, \\ (1, 0.05, 0), & 0.6 \leq p \leq 1. \end{cases} \tag{7}$$

The geometry coefficient jumps at $p = 0.2$; the formula is intentionally not smoothed. Other defaults are seed 2609, sequence length 2,048, learning rate 3×10^{-4} and a nominal 6M-token budget. Baseline microbatch/accumulation settings are $(2, 8)$ and Grassmannian Alignment settings are $(1, 16)$, giving 32,768 tokens per optimizer update in both cases. Token budgets are rounded to complete optimizer updates. AdamW is the default, with an optional AdEMAMix path; the learning-rate multiplier warms up over 10%, stays constant to 80%, then decays to 0.1. These settings describe the available script, not a recovered manifest of the September 18 evaluated runs.

B Training Algorithm

Algorithm 1 describes the language model implementation, with batch and layer averages explicit. It is not the spherical ASR training loop. The script computes geometric features even when their curriculum weight has decayed to zero; it restores the projection constraints every 50 optimizer steps rather than after each microbatch.

Algorithm 1 Grassmannian Alignment: language model distillation

Input: frozen teacher f_t; trainable student f_s; matched layer pairs $\mathcal{I}$; rank k; accumulation count a; token budget N rounded to complete optimizer updates.

1. Initialize each rectangular lift W_i with orthonormal columns; set token count $n = 0$, optimizer step $q = 0$ and accumulation count $c = 0$.
2. For each token microbatch X until the budget or time limit is reached at a complete optimizer update:
3. Evaluate $f_t(X)$ without gradients and $f_s(X)$ with gradients.
4. Compute next-token L_{CE} and token-mean L_{KL} at $\tau = 2$, with the τ^2 factor.
5. For each matched layer pair $i \in \mathcal{I}$: project $\widetilde{H}_{s,i} = W_i H_{s,i}$, average k contiguous temporal bins, and compute reduced QR bases $U_{s,i}, U_{t,i}$ in float32.
6. Average the layer and batch losses: $L_{\mathrm{G}} = |\mathcal{I}|^{-1} \sum_i \mathrm{mean}_b[1 - \left\|U_{s,i}^\top U_{t,i}\right\|_F^2 / k]$.
7. Obtain $w(n/N)$ from the curriculum in Appendix A; backpropagate $(w_{\mathrm{CE}} L_{\mathrm{CE}} + w_{\mathrm{KL}} L_{\mathrm{KL}} + w_{\mathrm{G}} L_{\mathrm{G}})/a$; increment c.
8. If $c = a$: clip the joint gradient norm to 1, update the student and lifts, advance the learning-rate schedule, clear gradients, reset $c = 0$, and increment q.
9. After that optimizer update, if q is divisible by 50, replace each W_i by its float32 QR Q factor, cast back to the training dtype.
10. Increment n by the number of tokens in X.

Output: student checkpoint; auxiliary lift and optimizer states for training resumption.

C Projector Identities and Their Scope

Proposition 1 (Basis invariance and principal angles)**.** *Let $U_s, U_t \in \mathbb{R}^{d\times k}$ have orthonormal columns. Their projector discrepancy satisfies Eq.* (2) *and is unchanged by independent right multiplication by orthogonal matrices.*

Proof. For $Q \in O(k)$, $(UQ)(UQ)^\top = UU^\top$. Since each rank-k orthogonal projector is symmetric and idempotent,

$$\|P_s - P_t\|_F^2 = \mathrm{tr}(P_s) + \mathrm{tr}(P_t) - 2\,\mathrm{tr}(P_s P_t) = 2k - 2\left\|U_s^\top U_t\right\|_F^2 .$$

The singular values of $U_s^\top U_t$ are $\cos\theta_j$, giving $2\sum_j \sin^2\theta_j$. These statements do not require choosing a unique QR basis. □

Proposition 2 (Average leakage and a bilinear bound)**.** *Let u be uniform on the unit sphere of the student subspace. Then*

$$\mathbb{E}\,\|(I - P_t)u\|_2^2 = \frac{L_{\mathrm{proj}}}{2k} = L_{\mathrm{G}}. \tag{8}$$

For any $x, y \in \mathbb{R}^d$,

$$|x^\top (P_s - P_t) y| \le \sqrt{L_{\mathrm{proj}}}\,\|x\|_2\,\|y\|_2 . \tag{9}$$

Proof. Rotational symmetry gives $\mathbb{E}[uu^\top] = P_s/k$. The first expression is therefore $\mathrm{tr}((I - P_t)P_s)/k = 1 - \left\|U_s^\top U_t\right\|_F^2 / k$. For the second expression, apply Cauchy–Schwarz, the operator-norm inequality, and $\|P_s - P_t\|_2 \le \|P_s - P_t\|_F$. □

If a *shared* downstream function f is C-Lipschitz, these identities also give $\|f(P_s x) - f(P_t x)\| \leq C\sqrt{L_{\text{proj}}}\,\|x\|_2$. The assumption of a shared function matters: this is not a guarantee for different teacher and student decoders, nor a WER or accuracy bound. It also does not prevent collapse to a common low-dimensional representation. The trace and rank assumptions must hold for the leakage interpretation to apply.

C.1 Projection and Geodesic Distances

The canonical Grassmann geodesic distance is $d_{\text{geo}} = (\sum_j \theta_j^2)^{1/2}$, whereas the implemented projector objective uses $L_{\text{proj}} = 2\sum_j \sin^2\theta_j$. Since $(2/\pi)\theta \leq \sin\theta \leq \theta$ on $[0, \pi/2]$,

$$\frac{8}{\pi^2} d_{\text{geo}}^2 \leq L_{\text{proj}} \leq 2 d_{\text{geo}}^2. \tag{10}$$

For small principal angles,

$$L_{\text{proj}} = 2d_{\text{geo}}^2 - \frac{2}{3}\sum_j \theta_j^4 + O\left(\sum_j \theta_j^6\right). \tag{11}$$

These losses are locally related but have different gradients. In rank one, $L_{\text{proj}} = 2\sin^2\theta$ and its angular derivative vanishes at $\theta = \pi/2$. The projector loss is bounded by $2k$, with equality requiring mutually orthogonal subspaces; this is impossible when $2k > d$. Zero loss means equal subspaces, not equal feature matrices or predictions.

C.2 Pullback Definiteness and Information Loss

For a smooth map ϕ, the Euclidean pullback satisfies

$$v^\top J_\phi^\top J_\phi v = \|J_\phi v\|_2^2. \tag{12}$$

It is positive definite precisely when the differential is injective on the source tangent space. For a full-column-rank linear map $\phi(x) = Wx$, $G = W^\top W$ is constant. Factoring $G = C^\top C$ with invertible C shows that $x \mapsto Cx$ is an isometry to Euclidean space. The induced source metric is therefore flat; this does not eliminate the separate Grassmann quotient geometry of subspaces in the target space.

If $A_s = 2A_t$ and A_t has rank k, their spans agree although their vectors differ. Projector matching consequently leaves scale and within-span coordinates unspecified. More generally, invertible right multiplication preserves the full column span of a matrix, but not necessarily a truncated leading singular subspace. Our code pools positions and applies fixed-rank QR rather than selecting leading singular vectors. Its basis invariance does not imply invariance to arbitrary changes in pooling, masking or rank.

C.3 Pooling and Shared-Lift Capacity

For fixed temporal averaging bins, linearity gives $\text{pool}_k(WH) = W\,\text{pool}_k(H)$. Applying the lift after pooling would therefore reduce its matrix-multiplication cost from $O(d_t d_s T)$ to $O(d_t d_s k)$, without changing the exact-arithmetic result. The implementation applies the lift before pooling.

Suppose both pooled matrices have full column rank k, and let P_W project orthogonally onto the range of a fixed lift W. Then $\text{range}(U_s) \subseteq \text{range}(W)$ and $P_W - P_s$ is positive semidefinite. Consequently,

$$L_{\text{G}} = 1 - \frac{\text{tr}(P_s P_t)}{k} \geq 1 - \frac{\text{tr}(P_W P_t)}{k} = \frac{\|(I - P_W)U_t\|_F^2}{k}. \tag{13}$$

For a distribution of sequences, write $\bar{P}_t = \mathbb{E}[P_t]$ and order its eigenvalues as $\lambda_1 \geq \cdots \geq \lambda_{d_t} \geq 0$. Since $\text{rank}(W) \leq d_s$, the variational characterization of leading eigenvalues yields

$$\mathbb{E}[L_{\text{G}}] \geq 1 - \frac{1}{k}\sum_{j=1}^{d_s} \lambda_j(\bar{P}_t). \tag{14}$$

This is a lower bound on the loss for a shared lift, not an attainable optimum for the parameterized student. Choosing a leading eigenspace for the lift maximizes the trace bound, but does not guarantee that the student realizes the optimal sequence-specific subspaces. The bound has not been measured for the evaluated checkpoints. At rank deficiency, arbitrary QR completion directions can violate the range inclusion used in the proof.

D Information Geometry and Optimization

D.1 Soft Targets and the High-Temperature Limit

The original distillation argument transfers relative probabilities among classes, including incorrect alternatives (Hinton et al., 2015). For C classes, define $p_t^{(\tau)} = \mathrm{softmax}(z_t/\tau)$ and $p_s^{(\tau)} = \mathrm{softmax}(z_s/\tau)$. Let $\bar{z} = z - C^{-1}(\mathbf{1}^\top z)\mathbf{1}$ center the logits. For fixed finite logits as $\tau \to \infty$,

$$p_j^{(\tau)} = \frac{1}{C} + \frac{\bar{z}_j}{C\tau} + O(\tau^{-2}), \qquad \tau^2 \mathrm{KL}(p_t^{(\tau)} \| p_s^{(\tau)}) = \frac{1}{2C}\|\bar{z}_t - \bar{z}_s\|_2^2 + O(\tau^{-1}). \tag{15}$$

This follows by expanding softmax around the uniform distribution and KL to second order. The centering removes the additive-logit ambiguity. The approximation requires large temperature relative to logit magnitudes; it is not an identity at the finite temperatures used in our experiments. Logit standardization addresses scale more directly (Sun et al., 2024), while adaptive distillation changes the intermediate target distribution (Shing et al., 2025). Neither operation is the same as matching hidden-state subspaces.

D.2 Fisher and Pullback Metrics

A regular identifiable statistical family $\{p(x \mid \theta)\}$ has Fisher information

$$F(\theta) = \mathbb{E}_{p_\theta}\left[\nabla_\theta \log p_\theta(X) \nabla_\theta \log p_\theta(X)^\top\right]. \tag{16}$$

Under differentiability and integrability conditions allowing expansion under the expectation,

$$\mathrm{KL}(p_\theta \| p_{\theta+\delta}) = \tfrac{1}{2}\delta^\top F(\theta)\delta + o(\|\delta\|^2). \tag{17}$$

Thus KL has a local metric interpretation, despite being asymmetric globally. Redundant parameterizations can make F singular; positive definiteness requires appropriate identifiability conditions (Amari & Nagaoka, 2000; Amari, 2016; Ay et al., 2017). This Fisher metric concerns probability distributions. The feature-space pullback $J_\phi^\top J_\phi$ instead pulls back a Euclidean target metric; they are not automatically the same tensor. Pulling back a non-Euclidean target metric would give $J_\phi^\top g_{\phi(x)} J_\phi$.

D.3 Curve Length, Geodesics and Implemented Updates

On a Riemannian manifold $(\mathcal{M}, g)$, a smooth curve has length

$$\mathrm{Length}(\gamma) = \int_a^b \sqrt{g_{\gamma(t)}(\dot{\gamma}(t), \dot{\gamma}(t))}\, dt. \tag{18}$$

An affinely parameterized geodesic satisfies

$$\ddot{x}^k + \sum_{i,j} \Gamma_{ij}^k \dot{x}^i \dot{x}^j = 0. \tag{19}$$

Geodesics are locally length minimizing on sufficiently short segments, not necessarily globally shortest over arbitrary intervals. A Riemannian gradient is defined through $g(\mathrm{grad}\, L, v) = dL[v]$. An intrinsic update may use an exponential map or a retraction,

$$x_{r+1} = \mathrm{Exp}_{x_r}(-\eta\, \mathrm{grad}\, L) \quad \text{or} \quad x_{r+1} = R_{x_r}(-\eta\, \mathrm{grad}\, L). \tag{20}$$

These are standard optimization constructions (Smith, 1994; Absil et al., 2008). Our LM script uses Euclidean optimizer steps and periodic QR restoration of the lift. It does not implement the geodesic equation, an exponential-map optimizer, or hyperbolic optimization. Normalizing Wx to unit length is also not, by itself, a spherical exponential map.

D.4 Hausdorff Separation and Representation Collapse

The Hausdorff property separates distinct points by disjoint neighborhoods (Hausdorff, 1914). Standard smooth manifolds, including the spheres and Grassmann manifolds considered here, are Hausdorff. This topological property does not require a neural representation map to be injective: a constant map into a Hausdorff space still collapses all inputs. It therefore cannot establish sample identifiability, prevent collapse, or guarantee optimization convergence. Rank conditions and task supervision remain necessary considerations in our formulation.

E Extended Experimental Details

E.1 Speech Training and Evaluation

The ASR student has 24 encoder and 24 decoder layers. Its rank-64 adapters target `q_proj`, `k_proj`, `v_proj`, `out_proj`, `fc1` and `fc2` in both stacks. The recorded adapter count is 69,206,016. A bias-free 1024×1280 lift adds 1,310,720 parameters, yielding 70,516,736 trainable parameters. The recorded 833,063,936 student total includes adapters but excludes this lift. The trainable fraction is approximately $70{,}516{,}736/(1.55 \times 10^9) = 4.55\%$ of the nominal teacher count.

Table 8: ASR parameter accounting from the saved model construction and printed counts. The teacher count is nominal; other counts are recorded or obtained by exact subtraction. Shading identifies our trainable and inference configurations. Ratios measure parameter counts, not memory, latency or recognition gains.

Component or configuration	Parameters
Whisper Large-v3 teacher (nominal)	1.55×10^9
Original Whisper Medium backbone	763,857,920
Rank-64 LoRA adapters	69,206,016
Training-only projection	1,310,720
Ours: initial trainable total (adapters + projection)	70,516,736
Ours: inference student (backbone + adapters)	833,063,936

The original backbone count is $833{,}063{,}936 - 69{,}206{,}016 = 763{,}857{,}920$. Relative to updating that entire backbone, the initial trainable total is 9.23%, a 90.77% reduction or $10.83\times$ fewer updated parameters. Relative to the nominal teacher, this total is 4.55% ($21.98\times$ fewer); the inference student retains 53.75% of the teacher's parameters (46.25% fewer, or $1.86\times$ smaller by count). Freezing the encoder at step 16,000 further restricts subsequent updates. The inference count conservatively includes unmerged adapters. No merged export, serialized-size comparison or matched peak-memory measurement is recorded. The geometric loss does not itself reduce parameter count: the smaller backbone provides model compression, and frozen-backbone LoRA provides adaptation savings.

Training interleaves German, French, Dutch, Spanish, Italian, Portuguese and Polish MLS streams (Pratap et al., 2020). Consumed audio hours are not logged; the availability of the 960-hour LibriSpeech training corpus (Panayotov et al., 2015) does not establish its consumption by this run. The main training-cell output identifies an H100 80GB HBM3 GPU. Other notebook variants identify L4 and T4 devices, but those records do not identify the hardware used for the reported recognition evaluation.

The main cell uses seed 42, float16 automatic mixed precision with gradient scaling, AdamW weight decay 0.01 and gradient clipping at norm 1.0. A cosine scheduler has $T_{\max} = 20{,}000$ and minimum learning rate 10^{-6}; stage boundaries overwrite the learning rate with the values in Table 7. At step 16,000 the encoder parameters are frozen. Every 200 steps, the code logs development loss on up to 50 utterances and saves a checkpoint. The "best" file is overwritten when the current single-utterance training loss improves the running minimum; development loss is not the selection criterion. Resumption initializes that minimum from the saved training-loss history. The last recorded H100 session resumes a completed run, and another cell writes to the same checkpoint filenames, so the hardware record does not establish the device used for every earlier update.

The checkpoint at step 17,600 loads with no missing or unexpected keys. Its stored final and minimum training losses are 1.6244 and 0.0623. These values are training metadata, not WER

measurements or ablation results. The evaluation streams LibriSpeech test-other with batch size one and English transcription until 200 utterances have been processed successfully. It skips empty normalized references and processing exceptions. Selected IDs, excluded IDs, hypotheses and reference transcripts are not retained in the supplied output.

After lowercasing, removing ASCII punctuation and collapsing whitespace, `jiwer` computes corpus-level WER and CER:

$$\mathrm{WER} = 100\frac{S_w + D_w + I_w}{N_w}, \qquad \mathrm{CER} = 100\frac{S_c + D_c + I_c}{N_c}. \tag{21}$$

Here S, D, I count substitutions, deletions and insertions, and N counts reference words or characters. The saved output reports 13.49% WER and 7.00% CER. The latter is a character error rate on this same subset, not a test-clean word error rate. Published Whisper reference scores in Table 2 use complete splits and the Whisper English normalizer (Gandhi et al., 2023). No matched noisy-speech evaluation or component ablation is available.

E.2 Language Model Evaluation Metadata

Table 9: Recorded LM evaluation configuration shared by both exports. These fields describe evaluation; they do not reconstruct training.

Field	Recorded value
Student parameters	438,349,352 each
Backend and precision	Hugging Face; bfloat16
Evaluator and Transformers	LM Evaluation Harness 0.4.13; Transformers 4.57.3
Device	`cuda:0`; NVIDIA GH200 120GB
Batch size and context	Automatic, resolved to 64; 4,096 tokens
Prompting	Zero-shot; no system instruction or chat template recorded
Evaluation seeds	Python 0; NumPy, PyTorch and few-shot sampler 1234
Bootstrap setting	100,000 iterations; not training repetitions
Checkpoint SHA and task hashes	Empty fields
Training manifest	Not included

Table 10: Evaluation splits and counts. XNLI and XCOPA counts are per language.

Task	Split	Available	Evaluated
ARC-Easy	test	2,376	1,000
ARC-Challenge	test	1,172	1,000
HellaSwag	validation	10,042	1,000
WinoGrande	validation	1,267	1,000
PIQA	validation	1,838	1,000
XNLI (each of 15 languages)	validation	2,490	1,000
XCOPA (each of 11 languages)	test	500	500

The XNLI codes in Table 5 are ar (Arabic), bg (Bulgarian), de (German), el (Greek), en (English), es (Spanish), fr (French), hi (Hindi), ru (Russian), sw (Swahili), th (Thai), tr (Turkish), ur (Urdu), vi (Vietnamese) and zh (Chinese). XCOPA additionally uses et (Estonian), ht (Haitian Creole), id (Indonesian), it (Italian), qu (Quechua) and ta (Tamil). Table 3 reports the raw metric; Table 14 retains every available length-normalized metric. Standard errors are those stored by the harness, and do not quantify variability across trained checkpoints.

For the reference training script, matched layer pairs lie near 25%, 50%, 75% and 100% of student depth, with proportionally selected teacher layers. The input pipeline, initialization, seed, consumed-token count and exact checkpoint identity must be recorded jointly to reproduce a run. The nominal FineWeb-Edu configuration is not evidence that the Swiss AI pretraining corpus or a 64-device training configuration was used. The code's full-student training also differs from the frozen-backbone ASR adaptation regime.

E.3 Implementation Checks and Controlled Comparisons

Algebraic checks should verify zero projector loss under $U \mapsto UQ$, agreement between explicit-projector and cross-basis formulas, and nonzero gradients on intended trainable modules away from stationary configurations. Numerical checks must detect non-finite QR outputs and rank deficiency. Such checks validate implementation properties; they do not establish downstream effectiveness. Controlled comparisons should hold data, updates, optimizer, curriculum and evaluation examples fixed, then vary the representation objective. Repeated seeds and paired predictions are needed to assess variability and example-level changes.

F Exploratory Image–Text Evaluation Records

The historical evaluation archive also contains per-example scores for ChartQA (Masry et al., 2022), DocVQA (Mathew et al., 2021), ScienceQA (Lu et al., 2022) and MMMU (Yue et al., 2024). These records do not identify the evaluated model or supply a matched baseline. We retain their verified aggregates for completeness, without attributing them to Grassmannian Alignment or extending the speech and language claims of the paper. The score sums are 1,836 for ChartQA, 4,253.4762996 for DocVQA, 3,544 for ScienceQA and 367 for the scored MMMU subset. DocVQA uses a continuous ANLS score; its sum is not a count of correct answers. Full inference and scoring cannot be reconstructed from these stored scores alone.

Table 11: Exploratory score records with unidentified model provenance. The last column is 100 times the mean non-null stored score, not a shared metric across tasks.

Task	Records	Scored	Missing	Score (%)
ChartQA	2,500	2,500	0	73.44
DocVQA (ANLS)	5,349	5,349	0	79.52
ScienceQA	4,241	4,241	0	83.57
MMMU	900	847	53	43.33

All 53 missing MMMU scores have non-null targets. For example, one architecture-and-engineering item has prediction "C." and target "1.06", while a basic-medical-science item has prediction "[H, a]." and target "C". The files do not identify whether these omissions arise from parsing, answer types or another scoring rule. Holding the recorded binary outcomes fixed and assigning each missing outcome $u_j \in \{0, 1\}$ gives

$$\frac{367}{900} \leq \frac{367 + \sum_{j=1}^{53} u_j}{900} \leq \frac{420}{900}, \tag{22}$$

or $[40.78\%, 46.67\%]$. This is a deterministic missing-outcome sensitivity interval, not a confidence interval or an observed complete-set score.

An image–text extension would require modality-specific tokenization, for example patch embeddings $v_1, \ldots, v_n$ concatenated with text embeddings $w_1, \ldots, w_m$, followed by an explicit teacher–student correspondence. The existence of such a token sequence does not establish a valid alignment or a completed distillation experiment. Training and evaluation of that extension remain outside the present claims.

G Further Context and Extensions

Contrastive predictive coding introduced InfoNCE as a predictive representation objective (Oord et al., 2018). Later analysis emphasizes the assumptions needed to recover latent factors and the gap between identifiability and downstream accuracy (Rusak et al., 2025). The ASR contrastive regularizer should therefore be interpreted as an auxiliary correspondence objective, not a guarantee of latent recovery. Distribution calibration for few-shot learning (Yang et al., 2021) transfers feature statistics across classes, which is distinct from matching a teacher's hidden-state subspace.

Scaling laws relate model size, data and computation to language-model loss (Kaplan et al., 2020); they do not justify a gain for a particular alignment loss. Hyperparameter benchmarks likewise

motivate controlled evaluation rather than a claim about geometric convergence (Klein & Hutter, 2019). Synthetic-data-assisted teacher compression (Liu et al., 2018a) changes the compression data distribution, an alternative direction that is not used in our recorded experiments.

Conditional computation through sparsely gated experts (Shazeer et al., 2017), Switch Transformers (Fedus et al., 2022) and Mixtral (Jiang et al., 2024) separates total capacity from active computation; Mu & Lin (2025) review this broader literature. Expert-based distillation for domain shift (Zhong et al., 2022) and expert adapters for continual vision–language learning (Yu et al., 2024) suggest possible future teacher or student configurations. They are not components of our dense-student implementation.

Representation-aware generative modeling includes visual autoregressive prediction (Tian et al., 2024), adapter-based classifier-free-guidance distillation (Perez Jensen & Sadat, 2025), and manifold-aware diffusion over encoder representations (Kumar & Patel, 2026). Their training targets differ from ours; geometric terminology alone does not imply equivalent objectives. Similarly, the Moonshine and Insanely Fast Whisper software projects (Moonshine AI, 2026; Vaibhavs10 and contributors, 2026) concern speech inference systems. They are external resources, not evidence that our checkpoint has been evaluated on an edge device.

H Additional ASR Training Records

The notebook also records a sequential loss-weight search and two short training runs with a Distil-Small.en student. These are training-loss records, not additional recognition evaluations or controlled ablations of the main Whisper Medium configuration.

H.1 Sequential Loss-Weight Search

The search covers 72 combinations of temperature $\tau \in \{3, 3.5, 4, 4.5, 5, 5.5\}$, KL weight $\{0.75, 1, 1.5, 2\}$ and spherical weight $\{0, 0.15, 0.3\}$. Each configuration receives 200 updates of a continuing run; the student is not reset between configurations. Table 12 summarizes the printed means.

Table 12: Recorded weighted training loss, averaged over six temperatures for each weight pair. These values compare objective scales and training stages, not WER.

w_{KL}	$w_{\mathrm{sph}} = 0$	$w_{\mathrm{sph}} = 0.15$	$w_{\mathrm{sph}} = 0.3$	Mean
0.75	68.0	67.0	67.2	67.4
1.00	85.7	86.7	87.7	86.7
1.50	126.3	126.7	125.9	126.3
2.00	165.9	166.9	165.9	166.2
Mean	111.5	111.8	111.7	111.7

The mean loss is strongly associated with the KL weight (Pearson $r = 0.997$ over the 72 recorded configurations). Ranking different weighted objectives by their magnitude cannot identify the best recognition model. The printed selection reports temperature 5, KL weight 0.75, spherical weight 0.15 and loss 37.3376; its printed 200-step mean is 68.7627. The lowest printed mean is 62.5850 at temperature 3 with the same weights. Accumulated search state and sequential training prevent interpreting this discrepancy as a validated optimum.

H.2 Short Training Comparisons

Two further cells run for approximately 30 minutes each. One combines CE, KL and a linear feature MSE; the other combines CE, KL, a temporal second-difference term and a sequence-level InfoNCE term. Both use batch size one and an English Distil-Small.en student on seven non-English MLS streams. Table 13 gives means over logged steps, not means over all updates.

Table 13: Windowed means of saved ASR training logs. Feature denotes linear MSE in the first run and the temporal term in the second. No recognition scores accompany these runs.

Objective	Logged steps	Count	CE	KL	Feature
Linear MSE	$[0, 500)$	10	9.57	143.0	0.542
	$[500, 1500)$	20	9.66	99.3	0.304
	$[1500, 3000)$	30	9.56	108.1	0.260
	$[3000, 4601)$	33	9.58	109.2	0.249
Temporal + InfoNCE	$[0, 500)$	10	9.46	144.9	0.003
	$[500, 1500)$	20	9.65	98.9	0.003
	$[1500, 3000)$	30	9.58	107.8	0.003
	$[3000, 4501)$	31	9.60	108.5	0.008

Logged CE averages remain between 9.4 and 9.7, while KL dominates the objective. The auxiliary sequence-level InfoNCE is identically zero at batch size one; the main cell instead contrasts frames. These runs do not isolate component effects in the evaluated model.

I Supplementary Language Model Evaluation

Table 14: Length-normalized accuracy (%), with reported standard errors and differences in percentage points. WinoGrande, XNLI and XCOPA do not have this metric in these exports.

Task	Baseline	Grassmannian Alignment	Δ
ARC-Easy	25.30 ± 1.38	26.70 ± 1.40	+1.40
ARC-Challenge	26.20 ± 1.39	23.50 ± 1.34	-2.70
HellaSwag	27.20 ± 1.41	27.90 ± 1.42	+0.70
PIQA	47.80 ± 1.58	47.50 ± 1.58	-0.30

Table 15: Complete XNLI scores and reported standard errors for our two runs.

Language	Baseline (ours)	Grassmannian Alignment (ours)	Δ
ar	33.30 ± 1.49	30.60 ± 1.46	-2.70
bg	34.20 ± 1.50	33.90 ± 1.50	-0.30
de	33.50 ± 1.49	34.00 ± 1.50	+0.50
el	33.10 ± 1.49	33.30 ± 1.49	+0.20
en	35.30 ± 1.51	31.60 ± 1.47	-3.70
es	33.30 ± 1.49	33.30 ± 1.49	+0.00
fr	33.30 ± 1.49	33.30 ± 1.49	+0.00
hi	33.10 ± 1.49	35.60 ± 1.51	+2.50
ru	32.10 ± 1.48	33.70 ± 1.50	+1.60
sw	33.80 ± 1.50	32.80 ± 1.49	-1.00
th	33.40 ± 1.49	33.60 ± 1.49	+0.20
tr	33.30 ± 1.49	33.30 ± 1.49	+0.00
ur	33.30 ± 1.49	33.30 ± 1.49	+0.00
vi	32.30 ± 1.48	33.40 ± 1.49	+1.10
zh	34.20 ± 1.50	31.80 ± 1.47	-2.40

Table 16: Complete XCOPA scores and reported standard errors for our two runs.

Language	Baseline (ours)	Grassmannian Alignment (ours)	Δ
et	49.40 $\pm$ 2.24	48.40 $\pm$ 2.24	-1.00
ht	52.60 $\pm$ 2.24	51.20 $\pm$ 2.24	-1.40
id	52.20 $\pm$ 2.24	52.20 $\pm$ 2.24	+0.00
it	51.00 $\pm$ 2.24	49.20 $\pm$ 2.24	-1.80
qu	50.40 $\pm$ 2.24	50.20 $\pm$ 2.24	-0.20
sw	51.40 $\pm$ 2.24	53.20 $\pm$ 2.23	+1.80
ta	57.00 $\pm$ 2.22	56.60 $\pm$ 2.22	-0.40
th	55.00 $\pm$ 2.23	54.00 $\pm$ 2.23	-1.00
tr	53.40 $\pm$ 2.23	50.80 $\pm$ 2.24	-2.60
vi	52.80 $\pm$ 2.23	52.00 $\pm$ 2.24	-0.80
zh	52.60 $\pm$ 2.24	52.00 $\pm$ 2.24	-0.60

I.1 Uncertainty and Controlled Follow-Up

Under an illustrative independent-binomial model, a two-sided test with $\alpha = 0.05$ and 80% power has approximate minimum detectable difference

$$\mathrm{MDE} \approx (1.96 + 0.84)\sqrt{2c(1-c)/n}, \tag{23}$$

where c is the nominal chance rate and n is the number of independent examples per model. For $n = 1{,}000$, this is 5.42 percentage points at $c = 0.25$ and 6.26 points at $c = 0.5$, exceeding the observed differences on the five English tasks. This planning calculation is not a paired significance test and does not establish equivalence. Both models were evaluated on the same examples, and translated multilingual items can be dependent; treating 15,000 XNLI or 5,500 XCOPA observations as independent would require additional justification. Paired predictions and repeated training seeds are unavailable.

A controlled study should initialize each arm from the same pretrained checkpoint or train a common initialization until benchmark performance clearly exceeds chance. Budgets of 10^8–10^9 tokens are candidate pilot extensions, not a guarantee of sufficient training. Hold data, optimizer, CE/KL schedule and evaluated examples fixed, toggle only the Grassmann term, and repeat across seeds. Report paired differences, uncertainty and training curves before drawing conclusions about geometry or convergence. The present comparison cannot distinguish limited budget from initialization, optimization, implementation or objective effects as explanations for weak performance.