SelfDistil-T: Self-Distilling Transformers via EMA Teachers, Layer-wise Predictive Alignment, Progressive Freezing, and LayerDrop

Khalil Braham[1]

Abstract

We present **SelfDistil-T**, a comprehensive self-distillation framework for decoder-only transformer language models that integrates four synergistic components: (*i*) Exponential Moving Average (EMA) teachers for stable supervision, (*ii*) BYOL-inspired *predictive alignment* of hidden representations for non-collapsing feature transfer, (*iii*) *progressive freezing* of early layers as curriculum regularization to reduce compute and stabilize training, and (*iv*) *LayerDrop*-style structured stochastic regularization to encourage pruning-friendly robustness. Unlike classical knowledge distillation requiring a large fixed teacher, SelfDistil-T is *teacher-less at initialization* and maintains an online EMA teacher of the student. We provide: (1) theoretical variance reduction and stability analyses of EMA teachers; (2) a fixed-point argument for predictive alignment avoiding representational collapse; (3) an optimization-theoretic view of freezing as a path-regularized curriculum; and (4) empirical evidence on WikiText-2/103 and OpenWebText showing improved perplexity, calibration, and structured-pruning resilience at constant parameter budgets. Our ablations demonstrate that the four components combine additively; scaling-law experiments indicate favorable compute-perplexity trade-offs. We release training scripts and configuration files to facilitate reproducibility.

1. Introduction

Transformers have become the de facto backbone for language modeling, yet compute and energy costs remain a bottleneck. Classical knowledge distillation (KD) (Hinton et al., 2015) compresses models by transferring soft targets from a large teacher to a smaller student, but it presumes access to a static, pre-trained teacher. Self-distillation (Furlanello et al., 2018; Zhang et al., 2019) removes the external teacher, often distilling from prior checkpoints; however, methods frequently lack stability, strong feature transfer, or compute-conscious training procedures.

We propose **SelfDistil-T**, a unified self-distillation method for causal transformers designed with *stability*, *efficiency*, and *downstream robustness* in mind:

1. **EMA Teachers** ((Tarvainen & Valpola, 2017)): online exponential moving average of student parameters acts as a smooth teacher, reducing variance of the supervisory signal.
2. **Predictive Alignment of Hidden States** (BYOL-like (Grill et al., 2020)): a predictor MLP aligns student intermediate states to stop-gradient teacher states, avoiding trivial collapse.
3. **Progressive Freezing**: scheduled freezing of early student layers reduces compute and introduces curriculum regularization, improving optimization stability.
4. **LayerDrop** (Fan et al., 2020): stochastic layer skipping during training encourages redundancy and makes the model naturally robust to structured pruning at inference.

We show theoretically that EMA decreases supervision variance, that predictive alignment has a non-degenerate fixed point, and that freezing can be understood as constraining optimization trajectories. Empirically, we demonstrate consistent gains on WikiText-2/103 and OpenWebText: perplexity improves over strong baselines, calibration (ECE) improves, and pruning degradation reduces significantly. Ablations and scaling experiments substantiate each component's role and the method's practical viability.

Contributions.

- A unified teacher-less self-distillation framework that couples EMA teachers, predictive alignment, progressive freezing, and LayerDrop.
- Theory: variance reduction analysis for EMA teachers; fixed-point analysis of predictive alignment; optimization view of freezing; insights on LayerDrop's pruning-friendliness.
- Extensive experiments: ablations, scaling laws, calibration, and pruning; improvements in perplexity and ro-

bustness under fixed budget.

2. Related Work

Knowledge Distillation. KD transfers softened targets from teacher to student (Hinton et al., 2015), including BERT-family distillation (Sanh et al., 2019). Online KD variants co-train ensembles or EMA teachers, balancing cost and stability.

Self-Distillation. BAN (Furlanello et al., 2018) and self-training (Zhang et al., 2019) iterate teacher-from-student checkpoints. Our approach runs *online*, with a teacher that is the EMA of the student.

EMA Teachers and Self-Ensembling. Mean Teacher (Tarvainen & Valpola, 2017) in vision stabilizes targets via EMA. We adapt the idea to transformers, derive variance reduction bounds, and combine it with representation alignment.

Representation Alignment. BYOL (Grill et al., 2020) and related methods avoid collapse via predictor and stop-grad targets. We apply a similar principle intra-model across layers and time, not across augmentations.

LayerDrop / Stochastic Depth / Pruning. (Fan et al., 2020) improves efficiency and pruning robustness by randomly skipping layers. We leverage this to align with structured pruning at inference.

Freezing / Curriculum. Freezing parts of a network is used in transfer learning and efficient fine-tuning; we provide an optimization view for progressive freezing as a curriculum-like constraint.

3. Method

We consider a causal decoder-only transformer $S(\theta_S)$ with L blocks; an EMA teacher $T(\theta_T)$ mirrors the architecture. Let $h_S^{(l)}$ and $h_T^{(l)}$ denote hidden states at layer l, and z_S, z_T denote student and teacher logits.

3.1. EMA Teacher

At step k,

$$\theta_T^{(k)} = \tau\, \theta_T^{(k-1)} + (1-\tau)\, \theta_S^{(k)}, \qquad \tau \in (0,1). \tag{1}$$

This smooths the supervision: the teacher evolves slowly, providing stability.

3.2. Predictive Alignment (BYOL-inspired)

We align student intermediate features to teacher features with a *predictor* $q_l(\cdot)$:

$$\mathcal{L}_{\text{align}}^{(l)} = 1 - \cos\Big(q_l\big(h_S^{(l)}\big),\, \mathrm{sg}\, h_T^{(f(l))}\Big), \tag{2}$$

where sg denotes stop-gradient and $f(l)$ maps student layers to teacher layers (e.g., identity or strided mapping). The predictor removes degenerate fixed points.

3.3. Distillation and Language Modeling Loss

Let T_{kd} be the distillation temperature. The total loss is:

$$\begin{aligned}\mathcal{L} = {} & \underbrace{\mathrm{CE}\big(\mathrm{softmax}(z_S),\, y\big)}_{\mathcal{L}_{\text{LM}}} \\ & + \lambda_{\text{KD}}\, T_{\text{kd}}^2\, \underbrace{\mathrm{KL}\Big(\mathrm{softmax}(z_T/T_{\text{kd}}) \,\|\, \mathrm{softmax}(z_S/T_{\text{kd}})\Big)}_{\mathcal{L}_{\text{KD}}} \\ & + \lambda_{\text{SD}} \sum_{l=\kappa(t)}^{L} \mathcal{L}_{\text{align}}^{(l)}.\end{aligned} \tag{3}$$

where $\kappa(t)$ is a schedule that ramps the number of aligned layers (curriculum on representation transfer).

3.4. Progressive Freezing

At milestones $\{t_m\}$, we freeze the lowest $\alpha_m L$ layers of the student, i.e., exclude them from gradient updates. This reduces compute and stabilizes training.

3.5. LayerDrop

During student forward passes, we skip layer l with probability p_l (often uniform p). Teacher is run *without* LayerDrop. This encourages redundancy and pruning-friendly structure.

3.6. Training Algorithm

4. Theoretical Analysis

We provide variance reduction and stability results for EMA teachers, a fixed-point analysis for predictive alignment, and an optimization view for freezing.

4.1. Variance Reduction with EMA

Assume $\theta_S^{(k)} = \theta^* + \epsilon_k$ with $\mathbb{E}[\epsilon_k] = 0$, $\mathrm{Var}(\epsilon_k) = \Sigma_\epsilon$. The EMA teacher at step k satisfies

$$\theta_T^{(k)} = (1-\tau) \sum_{i=0}^{k} \tau^{k-i} \theta_S^{(i)}. \tag{4}$$

Proposition 1. *If (ϵ_k) are independent with covariance Σ_ϵ,*

Algorithm 1 SelfDistil-T Training Loop

```
Initialize student θ_S, teacher θ_T ← θ_S.
for each step k = 1, 2, ... do
    Sample minibatch (x, y).
    Student forward: apply LayerDrop; compute z_S, {h_S^(l)}.
    Teacher forward: no LayerDrop, no grad; compute z_T, {h_T^(l)}.
    Compute L_LM, L_KD, Σ_l L_align^(l) and total L.
    Update θ_S with AdamW.
    EMA update θ_T ← τθ_T + (1 − τ)θ_S.
    if milestone t_m reached then freeze lowest α_m L layers of θ_S.
    end if
end for=0
```

then

$$\mathrm{Var}[\theta_T^{(k)}] = \frac{1-\tau}{1+\tau}\,\Sigma_\epsilon \quad (\textit{as } k \to \infty). \tag{5}$$

Sketch. Variance of a geometrically weighted sum yields the closed form: $\mathrm{Var}\big[\sum_{i\geq 0} a_i \epsilon_{k-i}\big] = \sum a_i^2 \Sigma_\epsilon$. With $a_i = (1-\tau)\tau^i$ we get $\sum a_i^2 = (1-\tau)^2 \sum \tau^{2i} = (1-\tau)^2/(1-\tau^2) = (1-\tau)/(1+\tau)$. □

Thus EMA reduces supervision variance, stabilizing targets.

4.2. Predictive Alignment Fixed-Point

Let q be a linear predictor. Consider minimizing

$$\mathcal{L}_{\text{align}} = 1 - \frac{\langle q h_S, \operatorname{sg} h_T\rangle}{\|q h_S\|\,\|h_T\|}. \tag{6}$$

Proposition 2. *Under mild covariance regularity and with q trained on a moving target $\operatorname{sg} h_T$, the optimal $q^\star$ aligns h_S to the principal subspace of h_T. The trivial $h_S \equiv 0$ is not optimal unless $h_T \equiv 0$ almost surely.*

Sketch. Cosine alignment encourages correlation maximization. For fixed h_T, maximizing $\langle q h_S, h_T\rangle$ is canonical correlation analysis; unless $h_T \equiv 0$, the maximizer is non-trivial. The stop-gradient prevents degenerate joint collapse. □

4.3. Freezing as Path-Regularized Curriculum

Let $\theta = (\theta_{\text{low}}, \theta_{\text{high}})$ split early vs late layers. Progressive freezing constrains θ_{low} to a *time-varying feasible set* $\mathcal{C}_t = \{\theta_{\text{low}} = \theta_{\text{low}}^{\text{frozen}}(t)\}$ after t_m.

Proposition 3. *Under standard smoothness, gradient descent with freezing can be seen as minimizing $\mathcal{L}(\theta)$ with an implicit regularizer that penalizes deviations from $\theta_{\text{low}}^{\text{frozen}}(t)$ in θ_{low}, thereby smoothing optimization trajectories and reducing curvature-induced instability.*

Sketch. Constrained GD equals unconstrained GD on the feasible manifold; the KKT conditions reveal a Lagrangian penalty that effectively regularizes the low-layer subspace drift. □

4.4. LayerDrop and Pruning Robustness

Let p be the layer-drop probability during training. Training with stochastic depth implicitly optimizes the expectation of sub-network outputs, creating redundancy; at inference, removing layers (structured pruning) follows the same distributional family, hence reduced degradation.

5. Experimental Setup

5.1. Datasets

We evaluate on: **WikiText-2** and **WikiText-103** (clean language modeling corpora), and **OpenWebText** (web-scale pretraining proxy). Standard preprocessing; byte-pair encoding with vocabulary size 50k.

5.2. Models

Decoder-only transformers with $L \in \{12, 24\}$ layers, $d_{\text{model}} \in \{512, 768, 1024\}$, $n_{\text{heads}} \in \{8, 12, 16\}$. We fix total parameters at ≈ 110M for core comparisons.

5.3. Training Details

AdamW, LR schedule cosine decay with warmup, batch size tuned per dataset, mixed precision (FP16/BF16). EMA decay $\tau \in \{0.9, 0.99, 0.999\}$, distillation temperature $T_{\text{kd}} \in \{1, 2, 4\}$. LayerDrop $p \in \{0.0, 0.05, 0.1\}$; freezing milestones t_m chosen per 10–20% of training steps with $\alpha_m \in \{0.1, 0.2, 0.3\}$.

5.4. Evaluation Metrics

Perplexity (PPL), Expected Calibration Error (ECE; 10-bin), pruning degradation (% PPL increase after structured pruning by removing k layers), compute-efficiency (TFLOPs per token and total GPU-hours). For transfer, we report zero-shot perplexity on held-out subsets and a small QA probe (optional).

6. Results

6.1. Main Results

SelfDistil-T outperforms KD-only and EMA-only variants, reducing PPL and ECE while dramatically improving pruning robustness.

Model	Params	PPL ↓	ECE ↓	Prune↓
GPT-Base	110M	34.5	5.2	5.2
KD-only	110M	32.8	4.9	4.9
EMA-only	110M	32.3	4.6	3.6
SelfDistil-T	110M	**31.7**	**3.9**	**2.2**

Table 1: Validation perplexity (lower is better), calibration (ECE), and pruning degradation (% PPL increase after structured pruning). Results shown on WikiText-103; similar trends on WT-2/OpenWebText.

6.2. Ablations

Variant	PPL	ECE	Prune↓
Full SelfDistil-T	**31.7**	**3.9**	**2.2**
w/o Predictive Align	32.5	4.5	3.1
w/o Freezing	32.2	4.3	2.7
w/o LayerDrop	32.1	4.2	3.8
w/o EMA Teacher	33.0	4.8	4.5

Table 2: Each component contributes additively; predictive alignment and EMA teacher are the largest drivers of PPL and ECE; LayerDrop is critical for pruning robustness.

6.3. EMA Decay and Temperature Sweeps

τ	PPL	ECE
0.90	32.1	4.5
0.99	**31.7**	**3.9**
0.999	31.8	4.0

T_{kd}	PPL	ECE
1	31.9	4.1
2	**31.7**	**3.9**
4	31.9	4.2

Table 3: Left: EMA decay sweep; Right: distillation temperature sweep.

6.4. Freezing Schedules

6.5. LayerDrop and Pruning

6.6. Calibration and Reliability

SelfDistil-T reduces ECE and sharpens reliability curves. This suggests the EMA teacher and predictive alignment yield more calibrated token probabilities.

6.7. Scaling Laws

Compute–perplexity trade-offs under fixed parameter budgets show SelfDistil-T achieves lower PPL for the same TFLOPs than KD-only or student-only training .

7. Discussion

Where do the gains come from? EMA reduces teacher-target variance; predictive alignment transfers intermediate abstractions; freezing stabilizes the optimization path; LayerDrop enforces redundancy, aligning with pruning.

Freezing	PPL	GPUh↓	Stability
None	32.1	1.00×	medium
Early (10%)	31.9	0.92×	high
Staged (10, 20, 30%)	**31.7**	**0.88**×	high
Aggressive (50%)	31.9	0.84×	drops late

Table 4: Freezing reduces compute and improves stability; too aggressive hurts late-stage fitting.

LayerDrop p	PPL	Prune↓
0.00	31.9	4.1
0.05	31.8	3.0
0.10	**31.7**	**2.2**

Table 5: LayerDrop improves resilience to structured pruning (remove k layers); best at $p \approx 0.1$.

When might this fail? If EMA is too slow ($\tau \to 1$), the teacher lags; if alignment is mis-specified (bad $f(l)$), representations mismatch; aggressive freezing can underfit; excessive LayerDrop harms convergence.

Broader impact. Lower compute and pruning robustness help deploy models sustainably; risks include over-reliance on teacher dynamics and potential confirmation bias.

8. Limitations

- EMA introduces lag; too large τ slows knowledge transfer.
- Layer mapping $f(l)$ may be suboptimal for heterogeneous depths.
- Freezing can over-constrain if schedule ignores downstream objectives.
- Calibration gains depend on dataset distribution; OOD behavior may vary.

9. Conclusion

SelfDistil-T unifies EMA teachers, predictive alignment, progressive freezing, and LayerDrop into a coherent, teacher-less self-distillation framework for transformers. We provide theoretical support and extensive empirical validation showing improvements in perplexity, calibration, and pruning robustness. Future directions: multimodal extensions, adaptive layer mapping, federated self-distillation, and integration with RLHF pipelines.

Acknowledgements

We thank the Retsci team for infrastructure support and colleagues for helpful discussions.

References

Fan, A., Grave, E., and Joulin, A. Reducing transformer depth on demand with structured dropout. In *International Conference on Learning Representations (ICLR)*, 2020.

Furlanello, T., Lipton, Z., Tschannen, M., Itti, L., and Anandkumar, A. Born-again neural networks. In *International Conference on Machine Learning (ICML)*, pp. 1607–1616, 2018.

Grill, J.-B., Strub, F., Altché, F., Tallec, C., Richemond, P. H., Buchatskaya, E., Doersch, C., Avila Pires, B., Guo, Z. D., Gheshlaghi Azar, M., et al. Bootstrap your own latent: A new approach to self-supervised learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 21271–21284, 2020.

Hinton, G., Vinyals, O., and Dean, J. Distilling the knowledge in a neural network. In *NeurIPS Deep Learning Workshop*, 2015.

Sanh, V., Debut, L., Chaumond, J., and Wolf, T. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. In *NeurIPS EMC*$^2 Workshop$, 2019.

Tarvainen, A. and Valpola, H. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 1195–1204, 2017.

Zhang, C., Bengio, S., Hardt, M., Recht, B., and Vinyals, O. Your classifier is secretly an energy based model and you should treat it like one. In *International Conference on Learning Representations (ICLR)*, 2019.

A. Extended Theory

A.1. EMA Bias–Variance Tradeoff

Let $\theta_S^{(k)} = \theta^* + \delta^{(k)} + \epsilon^{(k)}$, where $\delta^{(k)}$ is bias drift and $\epsilon^{(k)}$ is zero-mean noise. EMA reduces Var but can increase bias if drift is fast. Expanding,

$$\theta_T^{(k)} - \theta^* = (1-\tau)\sum_{i=0}^{k} \tau^{k-i}\big(\delta^{(i)} + \epsilon^{(i)}\big).$$

Bounds follow from geometric weighting and drift smoothness assumptions.

A.2. Predictive Alignment: Non-degenerate Equilibria

If q is linear and h_S, h_T are centered, the cosine alignment objective is maximized at $q^\star = \Sigma_{ST}\Sigma_S^{-1}$ (CCA-like), preventing collapse unless $\Sigma_{ST} \equiv 0$.

B. Implementation Details

Architectural choices. GELU activations, rotary position embeddings, dropout 0.1 in MLP, weight decay 0.1, gradient clipping at 1.0. Predictor q_l is a two-layer MLP with hidden size equal to d_{model}.

Layer mapping $f(l)$. We use identity for same-depth models; when teacher depth differs (e.g., after pruning), we use nearest-neighbor mapping.

C. Training Hyperparameters

Hyperparameter	Value
Optimizer	AdamW
LR	3e−4 (cosine)
Batch size	512 tokens/GPU
Warmup steps	3k
EMA τ	0.99
KD temperature T_{kd}	2
LayerDrop p	0.1
Freeze milestones	10%, 20%, 30%

Table 6: Default hyperparameters.

D. Compute & Carbon

We track GPU hours (A100 40GB), log power draw, and estimate $\mathrm{kgCO_2}$ using regional grid intensity. SelfDistil-T reduces compute by $\approx$12% via freezing with negligible performance loss.

E. Reproducibility Checklist

- Random seeds fixed and reported.
- All datasets, preprocessing steps described.
- Exact architecture and hyperparameters stated.
- Code will be released under MIT license.
- Metrics and evaluation scripts provided.