Spacing effect improves generalization in biological and artificial systems

Guanglong Sun[1,2,5,6], Ning Huang[1,2,6], Hongwei Yan[1,2,5], Liyuan Wang[3*], Jun Zhou[1,2], Qian Li[4], Bo Lei[5], and Yi Zhong[1,2]

[1]School of Life Sciences, IDG/McGovern Institute for Brain Research, Tsinghua University, Beijing, China
[2]Tsinghua-Peking Center for Life Sciences, Beijing, China
[3]Department of Psychological and Cognitive Sciences, Tsinghua University, Beijing, China
[4]School of Medicine, Shenzhen Campus of Sun Yat-Sen University, Shenzhen, Guangdong, China
[5]Beijing Academy of Artificial Intelligence, Beijing, China
[6]These authors contributed equally
[*]Correspondence: wly19@tsinghua.org.cn

SUMMARY

Generalization is a fundamental criterion for evaluating learning effectiveness, a domain where biological intelligence excels yet artificial intelligence continues to face challenges. In biological learning and memory, the well-documented *spacing effect* shows that appropriately spaced intervals between learning trials can significantly improve behavioral performance. While multiple theories have been proposed to explain its underlying mechanisms, one compelling hypothesis is that spaced training promotes integration of input and innate variations, thereby enhancing generalization to novel but related scenarios. Here we examine this hypothesis by introducing a bio-inspired spacing effect into artificial neural networks, integrating input and innate variations across spaced intervals at the neuronal, synaptic, and network levels. These spaced ensemble strategies yield significant performance gains across various benchmark datasets and network architectures. Biological experiments on *Drosophila* further validate the complementary effect of appropriate variations and spaced intervals in improving generalization, which together reveal a convergent computational principle shared by biological learning and machine learning.

KEYWORDS

Spacing Effect, Generalization, Ensemble Model, Bio-inspired Learning, NeuroAI

INTRODUCTION

The ability to generalize previously learned knowledge to novel but related scenarios is a hallmark of intelligent behavior. For example, a person who has learned to recognize other's voice can identify it again despite variations in background noise, pitch, or speaking speed. This underlies adaptive decision-making in everyday life and represents an almost innate advantage of biological intelligence (BI). In comparison, generalization from training data to similar test data is a fundamental objective of artificial intelligence (AI), especially for modern artificial neural networks (ANNs)-based systems[1]. Understanding the mechanisms that support robust generalization in BI offers a promising path toward improving generalization in AI. Conversely, implementing bio-inspired generalization strategies in ANNs allows for reverse-engineering and computational validation of candidate biological hypotheses. This interdisciplinary research paradigm, known as NeuroAI[1–3], holds promise for uncovering their convergent computational principles.

One well-documented phenomenon in biological learning and memory is the *spacing effect*[4]: spaced training with scattered repetitions over a certain period of time is significantly more effective than massed training with focused repetitions. This effect has been reported across species, from invertebrates to humans[5–12], and across a wide range of representative learning tasks[13–18]. Several cognitive theories have been proposed to account for this effect[19]. The *deficient-processing* theory[6,20–22] suggests that massed repetitions allow insufficient time for essential biological processes such as protein synthesis and synaptic plasticity, thereby impairing knowledge encoding. Alternatively, the *study-phase retrieval* theory[23,24] posits that spaced trials facilitate retrieval and strengthening of earlier memory traces, thereby enhancing knowledge retention. However, both theories are constrained to particular timescales: deficient-processing primarily explains short-term spacing effect (within an hour), whereas study-phase retrieval is more relevant to long-term spacing effect (hours to days). A more comprehensive explanation is provided by the *encoding variability* theory[25–28], which posits that spaced repetitions introduce natural variations in inputs or internal states, resulting in richer and more diverse memory traces. From this perspective, the role of spacing effect in improving generalization is achieved by appropriately integrating structured variability over time.

A parallel mechanism for improving generalization has emerged in machine learning, known as ensemble learning (EL)[3,29,30], which aggregates multiple models that capture diverse aspects of the data distribution. Similar to the temporally evolving nature of biological learning, temporal ensemble strategies[31,32] leverage variability accumulated across training time to construct more robust and generalizable representations (Fig. 1A,B), which include three representative implementations: (1) dropout[33] introduces stochasticity at the *neuronal level* by randomly deactivating units during training, promoting the formation of diverse activation patterns; (2) weight averaging (WA)[34], often implemented via exponential moving average (EMA)[35], aggregates parameter snapshots throughout network training, which captures variability at the *synaptic level*; (3) knowledge distillation (KD)[36], especially online KD[37] and self KD[38], transfers soft output distributions from temporally more advanced teacher models to student models, integrating variability at the *network level*.

Based on a unified formulation of temporal ensemble strategies, we revisit and computationally validate the encoding variability theory underlying spacing effect in biological systems. Specifically, we incorporate bio-inspired spacing effects into ANNs at the neuronal, synaptic, and network levels (corresponding to dropout[33], WA[34], and KD[36], respectively) by introducing controllable input and innate variations across spaced training intervals. Across a variety of benchmark datasets and network architectures, we observe a consistent U-shaped relationship of the training-to-test generalization with both the magnitude of variations and the duration of spaced intervals. Strikingly, either appropriate variation strength under fixed spaced interval or appropriate spaced interval under fixed variation strength leads to considerable improvements in test performance. These findings point to a critical balance between structured variability and temporal spacing for enhanced generalization.

Guided by these computational insights, we further design olfactory-conditioning experiments with *Drosophila* to assess whether biological generalization similarly benefits from this balance. We find that either increasing the spacing between repetitions or introducing variability in training cues significantly improves test performance in scenarios that differ slightly from training, suggesting that biological generalization also benefits from the structured integration of variations over a certain period of time. These findings demonstrate a shared computational principle underlying biological and artificial systems: structured exposure to appropriate variations promotes generalization from previous experiences. This principle opens up promising directions for designing biologically grounded training paradigms in ANNs, while offering a computational lens to understand naturally evolved “algorithms” in biological brains.

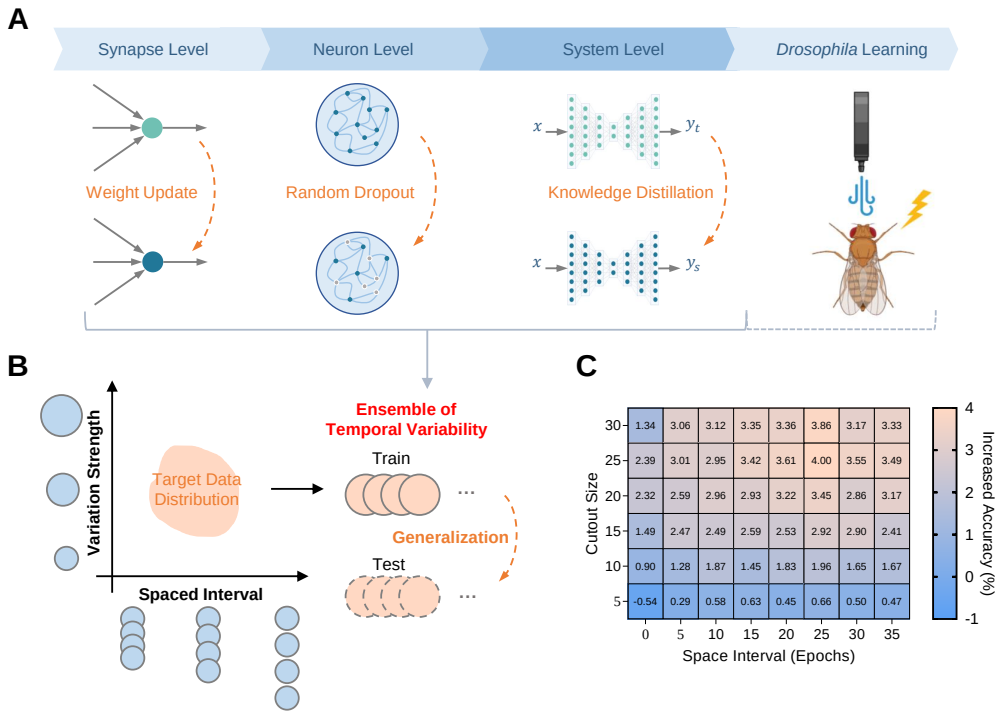

Figure 1: **Overview of spacing effect in biological and artificial systems.** (**A**) Schematic illustration of temporal ensemble strategies at the neuronal, synaptic, and network levels in ANNs, capturing the temporally evolving nature of biological learning. (**B**) Two key factors, variation strength and spaced interval, jointly modulate generalization performance in an independent and complementary manner. (**C**) Joint analysis of variation strength and spaced interval with cutout-based augmentation on CIFAR-10 dataset reveals a U-shaped performance profile with respect to both factors. All results are averaged over three runs with different random seeds.

RESULTS

Spaced temporal ensemble of encoding variability in biological and artificial systems

The encoding variability theory in BI posits that generalization is enhanced by appropriately spaced repetitions that introduce diverse representations, either through external inputs or internal neural states[6,25,26]. A parallel mechanism exists in AI, where EL improves generalization by aggregating appropriately diversified model variants that capture complementary aspects of the data distribution. Such diversity can arise from input variations introduced through external data augmentation[39] or from innate variations introduced throughout training. The innate variations are typically instantiated via temporal ensemble strategies such as dropout[33], WA[34], and KD[36], which operate at the neuronal, synaptic, and network levels, respectively.

In both biological and artificial systems, the effectiveness of variability in improving generalization critically depends on its magnitude and the temporal spacing over which it is integrated. This observation leads us to hypothesize a shared computational principle: structured exposure to appropriate variations over a certain period of time enhances generalization from training to test. To evaluate this, we systematically investigate how input and innate variations, modulated by controllable spaced intervals, affect generalization performance in ANNs, and further examine whether analogous effects occur in biological learning and memory through behavioral experiments in *Drosophila*.

Spacing effect with input and innate variations enhances generalization in ANNs

We begin by investigating the spacing effect in ANNs using externally imposed **input variations**. Specifically, we adopt the cutout augmentation strategy[40], which simulates environmental perturbations by randomly masking square patches in input images. This strategy allows independent manipulation of two key factors: the *variation strength* defined by the cutout size (Supplementary Fig. S1A), and the *spaced interval* defined by the frequency at which the cutout size is changed during training (Supplementary Fig. S1B).

Using a 4-layer convolutional neural network (CNN) trained on CIFAR-10 dataset[41], we first investigate how test performance varies with cutout size under a fixed augmentation schedule. We observe a clear U-shaped trend: moderate masking enhances generalization from training to test, whereas overly large occlusions impair the performance (Supplementary Fig. S1C). Next, we introduce temporal structure by periodically altering the cutout size across training epochs. Such spaced version of cutout augmentation further improves test performance and again yields a U-shaped trend, now with respect to the spaced interval (Supplementary Fig. S1D). Notably, the effects of variation strength and spaced interval appear largely independent and additive, suggesting that both factors contribute complementary benefits to generalization. When combined, they jointly promote more robust representations and improved test performance (Fig. 1C).

Recent theoretical work[42] has demonstrated that representative temporal ensemble strategies[31,32], including dropout[33], EMA[35], online KD[36], and self KD[38], are fundamentally equivalent in improving generalization, all amount to integrating properly diversified model variants throughout training (Fig. 1B). Building upon this foundation, we introduce a unified formulation that explicitly organizes temporal ensemble strategies under the common principle of spaced ensembling, governed by the *spaced interval* (s) between ensemble updates and the *variation strength* (k) applied at each update:

$$\mathcal{H}_t = \mathcal{F}\left(k \odot \mathcal{H}_{t-s}\right), \tag{1}$$

$\mathcal{H}_t$ denotes the internal state at the current training step t, which may refer to neuron activations h_t, network parameters θ_t, or model outputs $f_{\theta_t}(x)$ given inputs x. The operator $\odot$ represents element-wise modulation. $\mathcal{F}$ is an ensembling function adapted to each strategy. With this unified formulation, we empirically investigate the bio-inspired spacing effect in ANNs across three levels of innate variations.

Neuronal Level: Dropout Dropout[33] introduces innate variations by randomly deactivating a subset of neurons throughout training. Here, the *variation strength* corresponds to the dropout rate (i.e., the probability of deactivation), while the *spaced interval* is inherently fixed, as dropout is often applied at every training step. To explore the impact of temporal structure, we employ a periodic schedule on the dropout rate as the spaced version of dropout (Fig. 2A, Supplementary Algorithm 1), akin to the idea of implementing spacing effect with cutout augmentation. We observe a clear U-shaped relationship between test performance and dropout rate (Fig. 2B): mild deactivation promotes diverse activations and therefore improves generalization, whereas excessive sparsity of neuronal activations impairs the model's learning capacity. Importantly, this U-shaped trend persists when jointly manipulating the spaced interval s and the dropout rate as the variation strength k (Supplementary Fig. S2A), mirroring the spacing effect observed in input variations. The spaced version of dropout consistently outperforms standard dropout (Fig. 2C) across multiple benchmark datasets (CIFAR-100 and Tiny-ImageNet) and network architectures (ResNet-18, -50, and -101). The benefits generalize across advanced dropout variants including DropBlock[44], MaxDropout[43], and Checkerboard[45] (Fig. 2D). These results underscore the robustness of temporally structured neuronal variations in improving generalization.

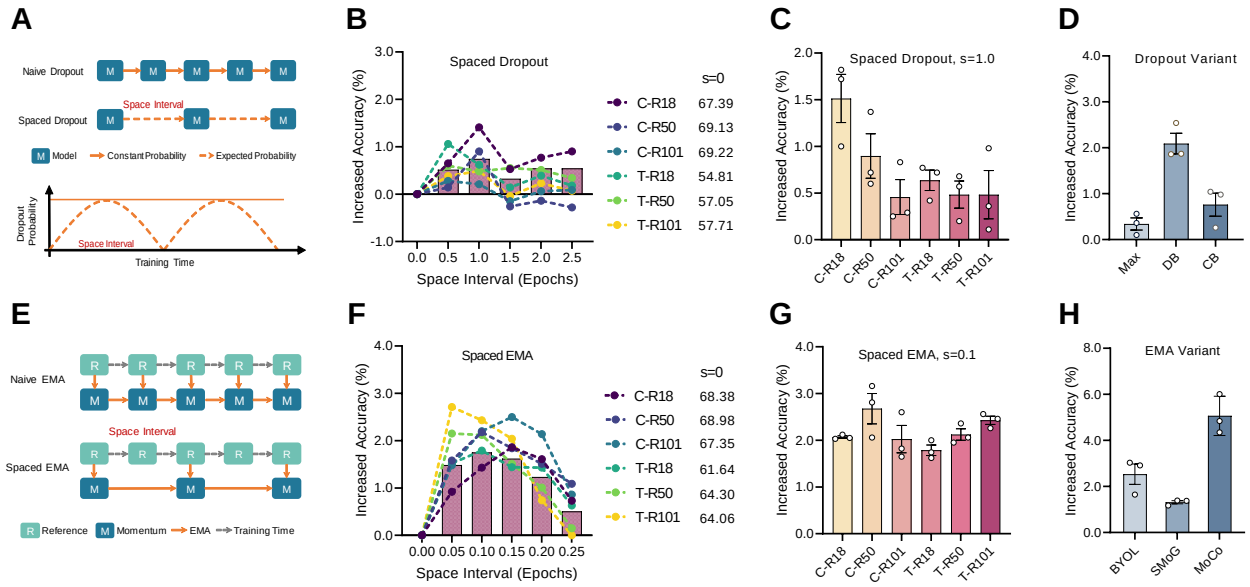

Figure 2: **Spacing effect enhances generalization through structured dropout and WA.** (**A**) Spaced version of dropout, where the dropout probability varies periodically to introduce structured neuronal variability during training. (**B**) Test accuracy under different structured dropout probabilities exhibits a U-shaped trend, indicating optimal performance at intermediate variation strengths. (**C**) Performance gains of spaced dropout across different network architectures and benchmark datasets. (**D**) Spaced venison of dropout improves performance in advanced dropout variants, including MaxDropout[43], DropBlock[44], and Checkerboard[45]. (**E**) Spaced version of EMA, where momentum updates occur at fixed intervals to control synaptic variability. (**F**) Test accuracy under different momentum spacing intervals also exhibits a U-shaped trend, highlighting the importance of optimal temporal spacing. (**G**) Performance gains of spaced EMA across different network architectures and benchmark datasets. (**H**) Spaced venison of EMA improves performance in advanced EMA variants with self-supervised learning, including BYOL[46], SMoG[47], and MoCo[48]. C, CIFAR-100. T, Tiny-ImageNet. R18, ResNet-18. R50, ResNet-50. R101, ResNet-101. All results are averaged over three runs with different random seeds. Data are presented as mean ± SEM.

Synaptic Level: Weight Averaging WA[34] such as EMA[35] enhances generalization by aggregating temporally spaced parameter snapshots throughout training. Here, the *spaced interval* is implicitly determined by the EMA momentum coefficient, which controls the effective temporal distance between aggregated snapshots (Fig. 2E, Supplementary Algorithm 2). The *variation strength* corresponds to the magnitude of parameter updates, which is not directly tunable in standard EMA. By modulating the momentum value, we again observe a clear U-shaped relationship between test performance and spaced interval (Fig. 2F): overly short intervals produce highly similar snapshots with limited diversity, while overly long intervals lead to incoherent aggregation that compromises performance. As shown in Supplementary Fig. S2B, different momentum values (serving here as the variation strength k) substantially affect the location of performance peak: lower momentum accelerates diversity but impairs stability, whereas higher momentum suppresses useful variations. The spaced version of EMA achieves consistent performance gains across multiple benchmark datasets and network architectures (Fig. 2G), and extend to more advanced EMA variants such as BYOL[46], SMoG[47] and MoCo[48] (Fig. 2G,H). These results validate the benefits of temporally spaced integration of encoding variability at the synaptic level.

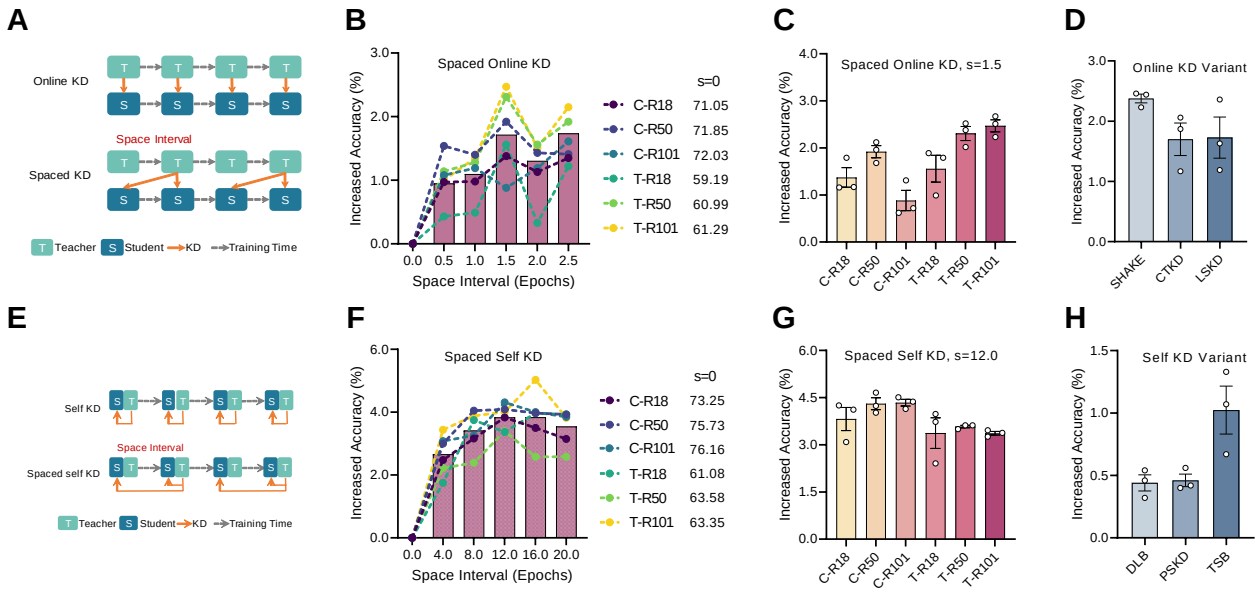

Figure 3: **Spacing effect enhances generalization in online KD and self KD.** (**A**) Spaced version of online KD, where the teacher model is updated at fixed intervals to induce innate variations. (**B**) Test accuracy of spaced online KD exhibits a U-shaped trend with respect to spaced interval. (**C**) Performance gains of spaced online KD across different network architectures and benchmark datasets. (**D**) Spaced venison of online KD improves performance in advanced online KD variants, including SHAKE[49], CTKD[50], and LSKD[51]. (**E**) Spaced version of self KD, where the deepest layers (teacher) periodically supervise shallower layers (students) with fixed intervals. (**F**) Test accuracy of spaced self KD exhibits a U-shaped trend with respect to spaced interval. (**G**) Performance gains of spaced self KD across different network architectures and benchmark datasets. (**H**) Spaced venison of self KD improves performance in advanced self KD variants, including DLB[52], PSKD[53] and TSB[54]. C, CIFAR-100. T, Tiny-ImageNet. R18, ResNet-18. R50, ResNet-50. R101, ResNet-101. All results are averaged over three runs with different random seeds. Data are presented as mean ± SEM.

Network Level: Knowledge Distillation KD[36] improves generalization by training student models with the soft outputs of more advanced teacher models. We investigate two representative temporal variants: online KD[37], where the teacher is asynchronously updated throughout training; and self KD[38], where the student periodically supervises itself using earlier snapshots. In both cases, the *spaced interval* corresponds to the number of epochs between teacher updates, while the *variation strength* naturally arises from the divergence between teacher and student over time. For online KD (Fig. 3A, Supplementary Algorithm 3), we observe a consistent U-shaped relationship between test performance and spaced interval (Fig. 3B): overly short intervals offer limited variations, while overly long intervals lead to teacher-student mismatch and degraded performance. A similar trend emerges in self KD (Fig. 3E–H, Supplementary Algorithm 4), where periodic re-alignment with earlier snapshots again exhibits a U-shaped performance profile (Fig. 3F). Further, Supplementary Fig. S2C confirms that adaptation of variation strength (e.g., via data shuffling) affects the optimal spaced interval, with stronger variations favoring shorter intervals. The spaced versions of online KD (Fig. 3C,D) and self KD (Fig. 3G,H) yield consistent performance gains across multiple benchmark datasets, network architectures, and advanced variants (e.g., SHAKE[49], CTKD[50], and LSKD[51] for online KD; and DLB[52], PSKD[53] and TSB[54] for self KD), suggesting the effectiveness of bio-inspired spacing effect at the network level.

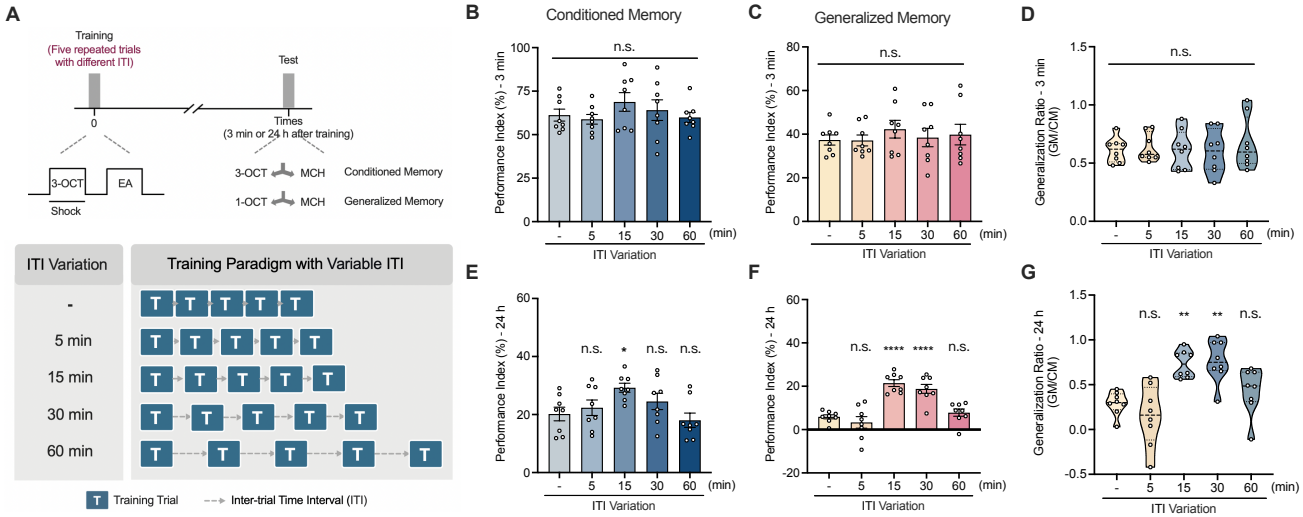

Figure 4: **Spaced training enhances memory generalization in *Drosophila*. (A)** Experimental paradigm for aversive olfactory conditioning with variable inter-trial intervals (ITI). The massed training group (-) receives five trials with a 45-s ITI, while the spaced training groups (5, 15, 30, 60 min) receive trials with increasing ITIs. Conditioned and generalized memory are tested by presenting the CS+ odor (3-OCT) or a structurally similar odor (1-OCT), respectively, against a novel odor (MCH). (**B**, **C**) Conditioned memory (**B**, n=8) and generalized memory (**C**, n=8) at 3 min show no significant difference across ITI groups. (**D**) The generalization ratio shows no significant difference across ITI groups at 3 min. n=8. (**E**, **F**) Conditioned memory (**E**, n=8) and generalized memory (**F**, n=8) at 24 h exhibit a U-shaped pattern across ITI groups. (**G**) The generalization ratio is significantly improved in the 15- and 30-min ITI groups at 24 h. n=8. Data are presented as mean ± SEM. Statistical significance is determined by one-way ANOVA with Dunnett's multiple comparisons test. *p=0.05, **p=0.01, ****p=0.0001, n.s., not significant.

Together, these results demonstrate that appropriately spaced exposure to either input or innate variations, whether introduced through data augmentation[40], dropout[33], WA[34], online KD[36], and self KD[38], consistently enhances generalization from training to test in ANNs. Moreover, Supplementary Fig. S2 highlights a consistent U-shaped relationship between variation strength k and spaced interval s across all methods, reinforcing the unified role of spacing-induced encoding variability from a machine learning perspective.

Spaced training and trial-to-trial variation enhance generalization in *Drosophila*

To investigate whether the unified role of spacing-induced encoding variability in ANNs also applies to biological systems, we train the *Drosophila* fruit flies using a classical olfactory aversive conditioning paradigm and assess both conditioned memory and its generalization performance[55–57] (see Supplementary Fig. S3A,B). In a forward training (FT) trial, flies are exposed to the odor 3-OCT paired with electric shocks (conditioned stimulus plus, CS+), followed by the odor EA without shock (conditioned stimulus minus, CS-). Both odor exposure only (OEO) trial and backward training (BT) trial serve as controls for evaluating associative memory performance. After training, conditioned and generalized memory are tested by presenting either the CS+ odor (3-OCT) or a structurally similar odor (1-OCT), respectively, against a novel odor (MCH). Our experiments validate that memory formation is associative and that flies can generalize to the similar odor as early as 3 min after training (Supplementary Fig. S3C-E).

First, we examine the effect of temporal spacing on memory generalization. We train flies with five repeated trials separated by different inter-trial intervals (ITIs), ranging from massed training

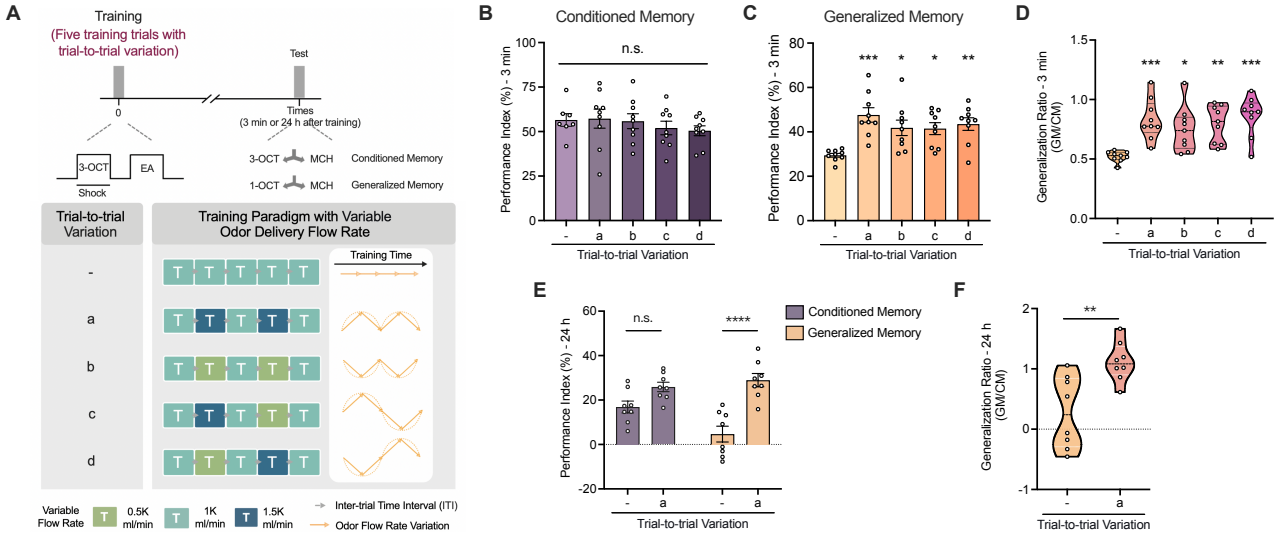

Figure 5: **Trial-to-trial variation during training enhances memory generalization in *Drosophila*.** (**A**) Experimental paradigm for massed training with variable sensory input. The control group (-) receives five trials with a constant odor delivery flow rate, while the experimental groups (a-d) receive trials with varying flow rates to introduce encoding variability. (**B**) Conditioned memory shows no significant difference between control and experimental groups (a-d) at 3 min. n=7-9. (**C**) All experimental groups (a-d) exhibit enhanced generalized memory at 3 min. n=9. (**D**) The generalization ratio is significantly increased in experimental groups (a-d) at 3 min. n=9. (**E**) Generalized memory remains enhanced in the experimental group (a) compared to the control group (-) at 24 h. n=8. (**F**) The generalization ratio remains elevated in the experimental group (a) at 24 h. n=8. Data are presented as mean ± SEM. Statistical significance is determined by one-way ANOVA with Dunnett's multiple comparisons test. *p=0.05, **p=0.01, ***p=0.001, ****p=0.0001, n.s., not significant.

(45-s ITI) to spaced training (5-, 15-, 30-, or 60-min ITI; Fig. 4A). At 3 min after training, all groups exhibit robust conditioned and generalized memory, with no significant differences across variable ITI conditions (Fig. 4B,C). The generalization ratio also has no significant difference among groups (Fig. 4D). In contrast, at 24 h after training, both conditioned and generalized memory are significantly influenced by ITI variation. Specifically, flies subjected to spaced training with 15- or 30-min ITIs show significantly enhanced conditioned and generalized memory compared to the massed training group (Fig. 4E,F), resulting in a U-shaped pattern across ITI groups (Fig. 4G). These results underscore the importance of spaced intervals in promoting memory generalization.

Next, we investigate the impact of encoding variability, inspired by our findings in machine learning. We introduce trial-to-trial variation during massed training by systematically varied the odor delivery flow rate (0.5, 1, or 1.5 K ml/min) across five trials (Fig. 5A). This manipulation does not impair conditioned memory, as all variable-training groups performed comparably to the constant-flow control group (Fig. 5B). Strikingly, introducing variability significantly enhance generalized memory (Fig. 5C) and increase the generalization ratio (Fig. 5D). This benefit is also persistent, as the enhancement in generalized memory is still evident 24 h after training (Fig. 5E,F). However, varying shock intensities across trials does not affect generalization under these conditions (Supplementary Fig. S4), indicating that variability in the conditioned sensory input (CS), rather than variability in the reinforcement strength (US), is the critical factor driving improved generalization.

Together, the biological findings in *Drosophila* are consistent with the computational results in

ANNs. Both appropriately spaced training intervals and increased trial-to-trial sensory variability can significantly and lastingly enhance the generalization of memories. These results demonstrate a convergent computational principle of biological and artificial systems, where structured exposure to temporal or contextual variations promotes the formation of more robust and generalizable representations.

DISCUSSION

In this study, we establish the encoding variability theory[25–27] of spacing effect as a shared computational principle that enhances generalization in both biological and artificial systems. Specifically, we demonstrate that appropriately spaced exposure to input or innate variations significantly improves generalization from training to test in ANNs (Fig. 2 and Fig. 3). This effect is robust across a variety of benchmark datasets, network architectures, and temporal ensemble strategies. Importantly, we find that the two key factors, variation strength and spaced interval, independently and additively enhance generalization. Beyond artificial systems, behavioral experiments in *Drosophila* reveal similar benefits under spaced exposure to variable training cues, showing that both spaced training intervals (Fig. 4) and increased sensory variations (Fig. 5) enhance memory generalization in olfactory associative learning. Together, these findings demonstrate how computational principles derived from artificial neural networks can guide biological experimentation, revealing that encoding variability operate as a unified mechanism that enhances generalization across both artificial and natural learners.

A central concept in our work is *generalization*. In machine learning, this typically refers to a model's ability to apply knowledge from training data to unseen test data drawn from the same underlying distribution[1]. In neuroscience, although animals are often tested with the same sensory cues presented during training, it is rarely possible to exactly reproduce the internal states or environmental conditions across training and testing episodes. As a result, what is traditionally studied as *precise memory* in neuroscience implicitly involves a form of "generalization"[3,58]. While the spacing effect has classically been linked to protein synthesis-dependent long-term memory, primarily enhancing conditioned memory, our findings provide, to our knowledge, the first demonstration that spaced training intrinsically and selectively promotes memory generalization. Unlike previously reported "rapid generalization" mechanisms[59], which rely on prior associative linking between environments, the generalization observed here appears to arise endogenously from the brain's natural processing of spaced experiences. Notably, this effect emerges after 24 hours but not at 3 minutes post-training (Fig. 4), suggesting that spaced learning facilitates the delayed integration and abstraction of acquired knowledge into new, related contexts, without requiring pre-configured associations. We suggest that this type of exact recall in biological learning (despite identical cues) is more analogous to the concept of generalization in machine learning (i.e., from training data to similar test data), while the broader notion of generalization in biological learning (e.g., transferring across different cues or contexts) parallels the more difficult domain generalization tasks in machine learning (i.e., from training data to dissimilar test data). Our work bridges this conceptual gap by showing how spacing and variability jointly support generalization in both fields.

The spacing effect has long been recognized in neuroscience for its beneficial impact on memory formation, but its computational implications for AI remain underexplored. Our results demonstrate that incorporating spacing mechanisms, particularly through temporal ensemble strategies that mimic the temporally evolving nature of biological learning, can substantially enhance test performance in ANNs. Unlike traditional EL methods that require training multiple models in parallel, temporal ensemble strategies efficiently aggregate internal variations over time. By further introducing structured spacing into these methods, we unlock new performance

gains while maintaining computational efficiency. These insights have practical implications for a range of AI challenges, including few-shot learning[60–62], continual learning[63–65], and adversarial robustness[66,67], where generalization from limited or noisy data is critical.

From the perspective of biological learning, prior studies have often manipulated either variation strength or spaced interval in isolation. For instance, encoding variability is typically studied by altering environmental conditions while keeping temporal spacing fixed, whereas spacing effects are evaluated by varying inter-trial intervals with constant stimuli[25–27]. Our work offers an integrated perspective by jointly manipulating both factors in controlled machine learning experiments. Inspired by temporal ensemble strategies[29–32], this dual modulation allows for a more comprehensive validation and extension of the encoding variability theory. The observed complementary effect between variation strength and spaced interval highlights their distinct, additive benefits to generalization, which may inform future designs of biological learning and memory experiments.

More broadly, this study exemplifies the promise of NeuroAI[1–3], an emerging field that seeks to accelerate progress in both neuroscience and AI through mutual inspiration. Our findings demonstrate how biologically grounded learning principles such as spacing and encoding variability can be translated into algorithmic gains for ANNs, while at the same time, controlled experiments in ANNs can generate testable hypotheses about biological learning mechanisms. Promising future directions include developing adaptive spacing schedules that dynamically adjust based on learning progress, extending multi-modal generalization across visual, auditory, and other sensory modalities, and integrating bio-inspired plasticity rules to further align ANNs with the dynamics of biological learning and memory. These directions may pave the way toward a unified and mechanistic understanding of generalization across both biological and artificial systems.

METHODS

Spacing Effect of Temporal Ensemble

We investigate how spacing effect in biological learning can be computationally leveraged to enhance generalization in ANNs via temporal ensemble strategies. Specifically, we instantiate this principle in three representative methods, WA, dropout, and KD, which operate at the synaptic, neuronal, and network levels, respectively. Our central hypothesis is that the generalization benefits observed in biological systems through spaced repetitions and encoding variability have computational analogues in ANNs, wherein variation strength and spaced interval jointly shape learning outcomes.

Problem Formulation

Let $\mathcal{D}_{\text{train}} = \{(x_i, y_i)\}_{i=1}^{N_{\text{train}}}$ and $\mathcal{D}_{\text{test}} = \{(x_i, y_i)\}_{i=1}^{N_{\text{test}}}$ denote the training and test sets drawn from the same underlying distribution, where each sample consists of an input-label pair. A neural network model $f_\theta(x)$ parameterized by θ is trained to minimize a supervised loss, defaulting to the cross-entropy (CE) loss for classification tasks:

$$\mathcal{L}(\theta; x, y) = -\sum_{c=1}^{C} y_c \log(f_\theta(x)_c), \tag{2}$$

where $f_\theta(x)_c$ is the predicted probability for class c, and y_c is a one-hot encoded ground-truth label.

The model's generalization performance $\mathcal{G}$ is evaluated on $\mathcal{D}_{\text{test}}$ using the average top-1 accuracy:

$$\mathcal{G}(\theta; \mathcal{D}_{\text{test}}) = \frac{1}{N_{\text{test}}} \sum_{i=1}^{N_{\text{test}}} \mathbb{I}\left[\arg\max_c f_\theta(x_i)_c = y_i\right], \tag{3}$$

where $\mathbb{I}[\cdot]$ is the indicator function, returning 1 if the predicted class matches the true label, and 0 otherwise. $f_\theta(x_i)_c$ denotes the predicted probability for class c on test input x_i.

Unified Temporal Ensemble Framework

Building upon prior theoretical efforts on temporal ensemble strategies[31,32,42], we introduce a unified formulation that captures the core components of spacing effect with two variables: *spaced interval* s and *variation strength* k. The ensemble update process can be summarized as:

$$\mathcal{H}_t = \mathcal{F}\left(k \odot \mathcal{H}_{t-s}\right), \tag{4}$$

where $\mathcal{H}_t$ denotes the internal state at the current training step t, which may refer to neuron activations h_t, network parameters θ_t, or model outputs $f_{\theta_t}(x)$ given input x. The operator $\odot$ represents element-wise modulation. $\mathcal{F}$ is an ensembling function adapted to each strategy. The following sections define these instances and describe how the bio-inspired spacing effect is implemented in each case.

Dropout Dropout[33] operates at the neuronal level by introducing stochastic perturbations to the neuronal activations. Let $\hat{h}_t$ denote the pre-activation vector at training step t. In the standard dropout formulation, a binary dropout mask $r_t \sim \text{Bernoulli}(p_t)$ is sampled element-wise with dropout rate p_t, and applied as:

$$h_t = r_t \odot \hat{h}_t, \tag{5}$$

where h_t denotes the post-dropout neuronal activations and $\odot$ denotes element-wise multiplication.

In our unified framework, the internal state at training step t is defined as $\mathcal{H}_t = h_t$, with temporal ensembling introduced throughout training:

$$\mathcal{H}_t = \mathcal{F}\left(k \odot h_{t-s}\right), \tag{6}$$

where s is the spaced interval, k corresponds to the dropout mask r_{t-s} from a previous training step, and $\mathcal{F}(\cdot)$ is a strategy-specific ensembling function (e.g., averaging or consistency regularization).

To further exploit the bio-inspired spacing effect, we modulate the dropout rate p_t periodically throughout training (Supplementary Algorithm 1). Specifically, we define:

$$p_t = p_{\text{max}} \cdot \sin\left(\frac{2\pi t}{s}\right), \tag{7}$$

where p_{max} is the maximum dropout rate, and s controls the period of modulation. This introduces structured variability in the dropout-induced representations, enabling both temporal spacing and controlled variation strength to improve generalization.

Weight Averaging WA[34] operates at the synaptic level by aggregating parameter snapshots throughout training. Let $\hat{\theta}_t$ denote the original parameters updated by gradient descent at training step t, and θ_t represent the ensembled parameters. In the standard EMA formulation, the ensemble is updated iteratively as:

$$\theta_t = \alpha\theta_{t-1} + (1-\alpha)\hat{\theta}_t, \tag{8}$$

where $\alpha \in [0, 1]$ is the momentum coefficient controlling the influence of past parameters. The underlying base parameters evolve via standard gradient descent:

$$\hat{\theta}_t = \hat{\theta}_{t-1} - \eta\nabla\mathcal{L}_{\text{task}}\left(f_{\hat{\theta}_t}(x), y\right), \tag{9}$$

with learning rate η and task loss $\mathcal{L}_{\text{task}}$.

In our unified framework, the internal state is defined as $\mathcal{H}_t = \theta_t$, and the ensembling process with spacing becomes:

$$\mathcal{H}_t = \mathcal{F}\left(k \odot \theta_{t-s}\right), \tag{10}$$

where s is the spacing interval, k corresponds to the momentum coefficient, and $\mathcal{F}(\cdot)$ denotes the EMA update.

To incorporate the spacing effect, we propose a spaced version of EMA (Supplementary Algorithm 2) in which parameter aggregation occurs every s steps:

$$\theta_t = \begin{cases} \alpha\theta_{t-s} + (1-\alpha)\hat{\theta}_t, & \text{if } t \bmod s = 0, \\ \theta_{t-s}, & \text{otherwise.} \end{cases} \tag{11}$$

This modification introduces explicit temporal spacing into the ensemble dynamics, allowing the model to integrate more diverse parameter states and improving robustness via spaced accumulation.

Knowledge Distillation KD[36] operates at the network level by guiding a student model with soft outputs of more advanced teacher models. Among commonly-used KD methods, online KD and self KD naturally align with the concept of temporal ensembling and are suitable for investigating the bio-inspired spacing effect.

Let $f_{\theta_t}(x)$ denote the teacher's output at training step t for input x, and $f_{\theta_{t-s}}(x)$ denote the student's output from an earlier checkpoint. In standard online KD, the teacher is a more advanced model (e.g., a later snapshot or moving average), and the student attempts to match the teacher's softened outputs. The distillation loss is defined as:

$$\mathcal{L}_{\text{KD}} = \text{KL}\left(f\theta_{t-s}(x) \parallel f_{\theta_t}(x)\right), \tag{12}$$

where KL$(\cdot \parallel \cdot)$ denotes the Kullback-Leibler divergence between teacher and student outputs. The student is trained using a combination of task and distillation losses:

$$\mathcal{L}_{\text{student}} = \mathcal{L}_{\text{task}}(f_{\theta_{t-s}}(x), y) + \lambda \mathcal{L}_{\text{KD}}, \tag{13}$$

where λ is a weighting coefficient balancing task supervision and distillation guidance.

In our unified framework, we define the internal state at time step $t-s$ as $\mathcal{H}_{t-s} = f_{\theta_{t-s}}(x)$. The spaced version of online KD introduces temporal ensembling through delayed teacher-student supervision:

$$\mathcal{H}_{t-s} = \mathcal{F}\left(k \odot f_{\theta_t}(x)\right), \tag{14}$$

where s is the spaced interval, k corresponds to a variation operator modulating the teacher signal (e.g., temperature scaling or input perturbation), and $\mathcal{F}(\cdot)$ is the distillation function (e.g., KL divergence or consistency regularization).

To enforce spacing explicitly, the teacher is updated every s steps based on task supervision:

$$\mathcal{L}_{\text{teacher}} = \mathcal{L}_{\text{task}}(f_{\theta_t}(x), y). \tag{15}$$

This temporal offset between teacher and student promotes representational diversity and prevents premature convergence to local minima.

For self KD[38], the teacher and student reside within the same network, typically across different layers, heads, or temporal branches. The spaced version of self KD (Supplementary Algorithm 4) alternates which internal representations serve as distillation targets over time. The formulation remains consistent in our framework, with spacing applied to internal states rather than model snapshots, allowing earlier components to imitate more refined representations from future computation stages.

Experimental Setup

To evaluate the impact of bio-inspired spacing effect on learning and generalization, we design parallel experiments in both biological and artificial systems.

Experiments with ANNs

To simulate input variability analogous to environmental changes in biological experiments, we first employ cutout augmentation[40] on a 4-layer CNN architecture. We then extend our study by incorporating the spacing effect into representative temporal ensemble strategies across synaptic, neuronal, and network levels.

Datasets and Training Protocols We evaluate on three benchmark datasets commonly used in image classification tasks, CIFAR-10[41], CIFAR-100[41], and Tiny-ImageNet[68]. CIFAR-10 contains 10 classes with 50k training and 10k testing images of size 32×32. CIFAR-100 contains 100 classes with 50k training and 10k testing images of size 32×32. Tiny-ImageNet contains 200 classes with 100k training and 10k testing images of size 64×64. Experiments are conducted on multiple backbone architectures, including ResNet-18, ResNet-50, and ResNet-101[69]. To ensure fair comparisons, we adopt identical training protocols following prior work[38,40,70]. Unless otherwise specified, all models are trained for 80 epochs using stochastic gradient descent (SGD) with momentum 0.9, batch size 128, and a constant learning rate 0.01.

Weight Averaging In standard EMA[35], the momentum model f_{θ_t} is updated at each training step from the reference model $f_{\hat{\theta}_t}$. The spaced version of EMA updates f_{θ_t} only every s steps, explicitly enforcing temporal spacing. The momentum coefficient α is set to 0.99 by default. We further implement spaced version of EMA within advanced self-supervised learning methods, including BYOL[46], SMoG[47] and MoCo[48], with the official codebase `https://github.com/lightly-ai/lightly`. In these methods, $f_{\hat{\theta}_t}$ is updated using contrastive loss, and f_{θ_t} is updated using EMA.

Dropout The standard dropout[33] introduces stochasticity by randomly zeroing out neuronal activations at each training step. The spaced version of dropout instead modulates the dropout probability periodically using a sinusoidal function parameterized by the spaced interval s and the variation strength k (i.e., the maximum dropout probability p_{max}). We set $p_{\text{max}} = 0.3$ by default. We further implement spaced version of dropout with advanced dropout variants, including DropBlock[44], MaxDropout[43], and Checkerboard Dropout[45].

Knowledge Distillation In standard online KD[37,71,72], the teacher model transfers knowledge to the student model at each training step. The spaced version of online KD introduces a spaced interval s (in epochs), such that the teacher is trained for s steps before guiding the student for the next s steps. In the case of self KD, where deeper layers act as teachers to guide shallower ones at each training step, the spaced version alternates between training the entire network for s steps and then training the shallow layers for s steps using the deepest layers' outputs. Following prior implementation[38,70], we set the KD temperature to 3.0, the feature loss coefficient to 0.03, and the distillation loss coefficient to 0.3. We further implement spaced version of more advanced online KD (SHAKE[49], CTKD[50], LSKD[51]) and self KD (DLB[52], PSKD[53], TSB[54]) variants.

Experiments with Biological Systems

This section describes the biological experimental setup used to examine the spacing effect in *Drosophila* melanogaster.

Fly Strains All fly stains are reared on standard cornmeal medium under a 12:12 hour light/dark cycle at 23°C and 60% humidity. We use wild-type flies of the the *W1118* strain, which are aged 2–3 days post-eclosion and included a mixture of males and females for behavioral assays.

Olfactory Aversive Conditioning To evaluate associative learning, we perform a classical Pavlovian olfactory conditioning procedure[55–57]. Flies are transferred to a behavioral room for at least 30 min to adapt to the experimental environment. Approximately 80-100 flies are subjected

to the following sequential stimuli in a forward conditioning trial: air for 90 s, an odor paired with 12 pulses of 60 V electric shock (CS+) for 1 min, air for 45 s, a second odor without pairing the electric shock (CS-) for 1 min, and finally air for 45 s. For backwards training, the electric shock ends 1 min before the onset of the first CS+ odor. In the odor exposure only (OEO) control, flies are exposed to both odors in the same sequence but without any electric shock delivery. All behavioral experiments are conducted at 25°C and 60% relative humidity. The following odorants are used, all diluted in mineral oil, including 4-methylcyclohexanol (MCH, 1.0×10^{-3} dilution, Fluka), 3-octanol (3-OCT, 1.5×10^{-3} dilution, Sigma-Aldrich), 1-octanol (1-OCT, 2.0×10^{-3} dilution, J&K Scientific), and ethyl acetate (EA, 1.0×10^{-3} dilution, Alfa Aesar).

Memory Testing and Performance Index Memory performance is evaluated using a standard T-maze apparatus at 3 minutes (immediate memory) or 24 hours (long-term memory) after training. Flies are given 1 minute to choose between the CS+ odor (or a structurally similar analog) and a novel odor. The preliminary performance index (PI) is calculated as:

$$\text{PI} = \frac{N_{\text{novel}} - N_{\text{CS+}}}{N_{\text{novel}} + N_{\text{CS+}}}, \tag{16}$$

where N_{novel} and $N_{\text{CS+}}$ denote the number of flies in the novel odor and CS+ arms, respectively. A PI of 100 indicated that all flies make the right choice to avoid the odor paired with the electric shock, while a PI of 0 indicates no memory retention, as reflected by a 50:50 distribution between the arms. To balance naive odor bias, two reciprocal groups are trained and tested simultaneously. One group is trained with CS+/CS-, and the other with odor exposure only or backwards training (i.e., shock delivered 1 minute before CS+ odor exposure). The complete PI is defined as the average PI of the two groups.

References

1. Ilievski, F., Hammer, B., van Harmelen, F., Paassen, B., Saralajew, S., Schmid, U., Biehl, M., Bolognesi, M., Dong, X.L., Gashteovski, K. et al. (2024). Aligning generalisation between humans and machines. arXiv preprint arXiv:2411.15626.

2. Zador, A., Escola, S., Richards, B., Ölveczky, B., Bengio, Y., Boahen, K., Botvinick, M., Chklovskii, D., Churchland, A., Clopath, C. et al. (2023). Catalyzing next-generation artificial intelligence through neuroai. Nature Communications *14*, 1597.

3. Wang, L., and Li, Q. (2025). Convergent multi-modular architecture for adaptive learning in drosophila and artificial intelligence. iScience.

4. Ebbinghaus, H. (2013). Memory: A contribution to experimental psychology. Annals of Neurosciences *20*, 155.

5. Beck, C., Schroeder, B., and Davis, R.L. (2000). Learning performance of normal and mutantdrosophila after repeated conditioning trials with discrete stimuli. Journal of Neuroscience *20*, 2944–2953.

6. Pagani, M.R., Oishi, K., Gelb, B.D., and Zhong, Y. (2009). The phosphatase shp2 regulates the spacing effect for long-term memory induction. Cell *139*, 186–198.

7. Menzel, R., Manz, G., Menzel, R., and Greggers, U. (2001). Massed and spaced learning in honeybees: the role of cs, us, the intertrial interval, and the test interval. Learning & Memory *8*, 198–208.

8. Anderson, M.J., Jablonski, S.A., and Klimas, D.B. (2008). Spaced initial stimulus familiarization enhances novelty preference in long-evans rats. Behavioural Processes *78*, 481–486.

9. Bello-Medina, P.C., Sánchez-Carrasco, L., González-Ornelas, N.R., Jeffery, K.J., and Ramírez-Amaya, V. (2013). Differential effects of spaced vs. massed training in long-term object-identity and object-location recognition memory. Behavioural Brain Research *250*, 102–113.

10. Medin, D.L. (1974). The comparative study of memory. Journal of Human Evolution *3*, 455–463.

11. Robbins, D., and Bush, C.T. (1973). Memory in great apes. Journal of Experimental Psychology *97*, 344.

12. Kukushkin, N.V., and Carew, T.J. (2017). Memory takes time. Neuron *95*, 259–279.

13. Donovan, J.J., and Radosevich, D.J. (1999). A meta-analytic review of the distribution of practice effect: Now you see it, now you don't. Journal of Applied Psychology *84*, 795.

14. Shea, C.H., Lai, Q., Black, C., and Park, J.H. (2000). Spacing practice sessions across days benefits the learning of motor skills. Human Movement Science *19*, 737–760.

15. Gluckman, M., Vlach, H.A., and Sandhofer, C.M. (2014). Spacing simultaneously promotes multiple forms of learning in children's science curriculum. Applied Cognitive Psychology *28*, 266–273.

16. Roediger, H.L., and Byrne, J. (2008). Learning and memory: A comprehensive reference (vol. 2). Cognitive Psychology of Memory.

17. Sobel, H.S., Cepeda, N.J., and Kapler, I.V. (2011). Spacing effects in real-world classroom vocabulary learning. Applied Cognitive Psychology *25*, 763–767.

18. Vlach, H.A. (2014). The spacing effect in children's generalization of knowledge: Allowing children time to forget promotes their ability to learn. Child Development Perspectives *8*, 163–168.

19. Smolen, P., Zhang, Y., and Byrne, J.H. (2016). The right time to learn: mechanisms and optimization of spaced learning. Nature Reviews Neuroscience *17*, 77–88.

20. Solso, R.L. (2024). Theories in cognitive psychology: The Loyola symposium. Taylor & Francis.

21. Toppino, T.C. (1991). The spacing effect in young children's free recall: Support for automatic-process explanations. Memory & cognition *19*, 159–167.

22. Philips, G.T., Ye, X., Kopec, A.M., and Carew, T.J. (2013). Mapk establishes a molecular context that defines effective training patterns for long-term memory formation. Journal of Neuroscience *33*, 7565–7573.

23. Hintzman, D.L., Summers, J.J., and Block, R.A. (1975). Spacing judgments as an index of study-phase retrieval. Journal of Experimental Psychology: Human Learning and Memory *1*, 31.

24. Tzeng, O.J., and Cotton, B. (1980). A study-phase retrieval model of temporal coding. Journal of Experimental Psychology: Human Learning and Memory *6*, 705.

25. Benjamin, A.S., and Tullis, J. (2010). What makes distributed practice effective? Cognitive Psychology *61*, 228–247.

26. Melton, A.W. (1970). The situation with respect to the spacing of repetitions and memory. Journal of Verbal Learning and Verbal Behavior *9*, 596–606.

27. Madigan, S.A. (1969). Intraserial repetition and coding processes in free recall. Journal of Verbal Learning and Verbal Behavior *8*, 828–835.

28. Kukushkin, N.V., Carney, R.E., Tabassum, T., and Carew, T.J. (2024). The massed-spaced learning effect in non-neural human cells. Nature communications *15*, 9635.

29. Yang, Y., Lv, H., and Chen, N. (2023). A survey on ensemble learning under the era of deep learning. Artificial Intelligence Review *56*, 5545–5589.

30. Ganaie, M.A., Hu, M., Malik, A.K., Tanveer, M., and Suganthan, P.N. (2022). Ensemble deep learning: A review. Engineering Applications of Artificial Intelligence *115*, 105151.

31. Laine, S., and Aila, T. (2016). Temporal ensembling for semi-supervised learning. arXiv preprint arXiv:1610.02242.

32. Hao, L., Healey, C.G., and Bass, S.A. (2015). Effective visualization of temporal ensembles. IEEE Transactions on Visualization and Computer Graphics *22*, 787–796.

33. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. Journal of Machine Learning Research *15*, 1929–1958.

34. Rame, A., Kirchmeyer, M., Rahier, T., Rakotomamonjy, A., Gallinari, P., and Cord, M. (2022). Diverse weight averaging for out-of-distribution generalization. Advances in Neural Information Processing Systems *35*, 10821–10836.

35. Polyak, B.T., and Juditsky, A.B. (1992). Acceleration of stochastic approximation by averaging. SIAM Journal on Control and Optimization *30*, 838–855.

36. Hinton, G., Vinyals, O., and Dean, J. (2015). Distilling the knowledge in a neural network. arXiv preprint arXiv:1503.02531.

37. Chen, D., Mei, J.P., Wang, C., Feng, Y., and Chen, C. (2020). Online knowledge distillation with diverse peers. In Proceedings of the AAAI Conference on Artificial Intelligence vol. 34. pp. 3430–3437.

38. Zhang, L., Song, J., Gao, A., Chen, J., Bao, C., and Ma, K. (2019). Be your own teacher: Improve the performance of convolutional neural networks via self distillation. In Proceedings of the IEEE/CVF International Conference on Computer Vision.

39. Mumuni, A., and Mumuni, F. (2022). Data augmentation: A comprehensive survey of modern approaches. Array *16*, 100258.

40. DeVries, T. (2017). Improved regularization of convolutional neural networks with cutout. arXiv preprint arXiv:1708.04552.

41. Krizhevsky, A., Hinton, G. et al. (2009). Learning multiple layers of features from tiny images.

42. Allen-Zhu, Z., and Li, Y. (). Towards understanding ensemble, knowledge distillation and self-distillation in deep learning. In International Conference on Learning Representations.

43. do Santos, C.F.G., Colombo, D., Roder, M., and Papa, J.P. (2021). Maxdropout: deep neural network regularization based on maximum output values. In International Conference on Pattern Recognition. IEEE pp. 2671–2676.

44. Ghiasi, G., Lin, T.Y., and Le, Q.V. (2018). Dropblock: A regularization method for convolutional networks. Advances in Neural Information Processing Systems *31*.

45. Nguyen, K.B., Choi, J., and Yang, J.S. (2022). Checkerboard dropout: A structured dropout with checkerboard pattern for convolutional neural networks. IEEE Access *10*, 76044–76054.

46. Grill, J.B., Strub, F., Altché, F., Tallec, C., Richemond, P., Buchatskaya, E., Doersch, C., Avila Pires, B., Guo, Z., Gheshlaghi Azar, M. et al. (2020). Bootstrap your own latent-a new approach to self-supervised learning. Advances in Neural Information Processing Systems *33*, 21271–21284.

47. Pang, B., Zhang, Y., Li, Y., Cai, J., and Lu, C. (2022). Unsupervised visual representation learning by synchronous momentum grouping. In European Conference on Computer Vision. Springer pp. 265–282.

48. He, K., Fan, H., Wu, Y., Xie, S., and Girshick, R. (2020). Momentum contrast for unsupervised visual representation learning. . URL: `https://arxiv.org/abs/1911.05722`. `arXiv:1911.05722`.

49. Li, L., and Jin, Z. (2022). Shadow knowledge distillation: Bridging offline and online knowledge transfer. Advances in Neural Information Processing Systems *35*, 635–649.

50. Li, Z., Li, X., Yang, L., Zhao, B., Song, R., Luo, L., Li, J., and Yang, J. (2023). Curriculum temperature for knowledge distillation. In Proceedings of the AAAI Conference on Artificial Intelligence vol. 37. pp. 1504–1512.

51. Sun, S., Ren, W., Li, J., Wang, R., and Cao, X. (2024). Logit standardization in knowledge distillation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 15731–15740.

52. Shen, Y., Xu, L., Yang, Y., Li, Y., and Guo, Y. (2022). Self-distillation from the last mini-batch for consistency regularization. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 11943–11952.

53. Kim, K., Ji, B., Yoon, D., and Hwang, S. (2021). Self-knowledge distillation with progressive refinement of targets. In Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 6567–6576.

54. Li, C., Wang, Z., and Qi, H. (2022). Online knowledge distillation by temporal-spatial boosting. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. pp. 197–206.

55. Tully, T., and Quinn, W.G. (1985). Classical conditioning and retention in normal and mutant drosophila melanogaster. Journal of Comparative Physiology A *157*, 263–277.

56. Tully, T., Preat, T., Boynton, S., and Del Vecchio, M. (1994). Genetic dissection of consolidated memory in drosophila. Cell *79*, 35–47.

57. Barth, J., Dipt, S., Pech, U., Hermann, M., Riemensperger, T., and Fiala, A. (2014). Differential associative training enhances olfactory acuity in drosophila melanogaster. Journal of Neuroscience *34*, 1819–1837.

58. Zhao, B., Sun, J., Zhang, X., Mo, H., Niu, Y., Li, Q., Wang, L., and Zhong, Y. (2019). Long-term memory is formed immediately without the need for protein synthesis-dependent consolidation in drosophila. Nature Communications *10*, 4550.

59. Zhou, H., Xiong, G.J., Jing, L., Song, N.N., Pu, D.L., Tang, X., He, X.B., Xu, F.Q., Huang, J.F., Li, L.J. et al. (2017). The interhemispheric ca1 circuit governs rapid generalisation but not fear memory. Nature Communications *8*, 2190.

60. Wang, Y., Yao, Q., Kwok, J.T., and Ni, L.M. (2020). Generalizing from a few examples: A survey on few-shot learning. ACM Computing Surveys *53*, 1–34.

61. Parnami, A., and Lee, M. (2022). Learning from few examples: A summary of approaches to few-shot learning. arXiv preprint arXiv:2203.04291.

62. Song, Y., Wang, T., Cai, P., Mondal, S.K., and Sahoo, J.P. (2023). A comprehensive survey of few-shot learning: Evolution, applications, challenges, and opportunities. ACM Computing Surveys *55*, 1–40.

63. Wang, L., Zhang, X., Su, H., and Zhu, J. (2024). A comprehensive survey of continual learning: Theory, method and application. IEEE Transactions on Pattern Analysis and Machine Intelligence *46*, 5362–5383.

64. De Lange, M., Aljundi, R., Masana, M., Parisot, S., Jia, X., Leonardis, A., Slabaugh, G., and Tuytelaars, T. (2021). A continual learning survey: Defying forgetting in classification tasks. IEEE Transactions on Pattern Analysis and Machine Intelligence *44*, 3366–3385.

65. Zhou, D.W., Sun, H.L., Ning, J., Ye, H.J., and Zhan, D.C. (2024). Continual learning with pre-trained models: A survey. arXiv preprint arXiv:2401.16386.

66. Stutz, D., Hein, M., and Schiele, B. (2019). Disentangling adversarial robustness and generalization. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 6976–6987.

67. Bai, T., Luo, J., Zhao, J., Wen, B., and Wang, Q. (2021). Recent advances in adversarial training for adversarial robustness. arXiv preprint arXiv:2102.01356.

68. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M. et al. (2015). Imagenet large scale visual recognition challenge. International Journal of Computer Vision *115*, 211–252.

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

70. Sun, G., Yan, H., Wang, L., Li, Q., Lei, B., and Zhong, Y. (2025). Right time to learn:promoting generalization via bio-inspired spacing effect in knowledge distillation. . URL: `https://arxiv.org/abs/2502.06192`. `arXiv:2502.06192`.

71. Zhang, Y., Xiang, T., Hospedales, T.M., and Lu, H. (2018). Deep mutual learning. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 4320–4328.

72. Zhu, X., Gong, S. et al. (2018). Knowledge distillation by on-the-fly native ensemble. Advances in Neural Information Processing Systems *31*.

RESOURCE AVAILABILITY

Lead contact

Requests for further information and resources should be directed to and will be fulfilled by the lead contact, Liyuan Wang (wly19@tsinghua.org.cn).

Materials availability

This study did not generate new materials.

Data and code availability

Data Availability

All benchmark datasets used in this paper are publicly available, including CIFAR-10/100[41] (`https://www.cs.toronto.edu/~kriz/cifar.html`) and Tiny-ImageNet[68] (`https://www.image-net.org/download.php`). Statistical analysis is performed using GraphPad Prism software. Data is considered normally distributed if they passed the Shapiro-Wilk test (for $n<8$) or the Anderson-Darling test (for $n\geq 8$). For normally distributed data, comparisons between the two groups are performed using the two-tailed unpaired t-test; comparisons between multiple groups are performed using the One-way ANOVA test followed by Dunnett's multiple comparisons test, and the two-way ANOVA test followed by Sidak's multiple comparisons test. Results are reported as n.s. (not significant) for $p > 0.05$, * for $p < 0.05$, ** for $p < 0.01$, *** for $p < 0.001$, and **** for $p < 0.0001$.

Code Availability

The implementation code is available at Zenodo `https://zenodo.org/records/17959984` and GitHub `https://github.com/SunGL001/spacing_generalization`.

ACKNOWLEDGMENTS

This work is supported by the STI2030-Major Projects (2022ZD0204900 to Y.Z.), the NSFC Projects (Nos. 62406160 to L.W., 32021002 to Y.Z.), and National Science and Technology Major Project (2022ZD01163013 to B.L.).

AUTHOR CONTRIBUTIONS

Study conception and design: G.S., L.W. Computational experiment performing: G.S., H.Y. Biological experiment performing: N.H. Visualization and data analysis: G.S., N.H., H.Y., L.W. Funding acquisition: Y.Z., L.W., B.L. Results were discussed and interpreted by: G.S., N.H., H.Y., L.W., J.Z., Q.L., B.L., Y.Z. Manuscript was written and discussed by: G.S., N.H., H.Y., L.W., J.Z., Q.L., B.L., Y.Z.

DECLARATION OF INTERESTS

The authors declare no competing interests.

DECLARATION OF GENERATIVE AI AND AI-ASSISTED TECHNOLOGIES

Large language models were used to polish the manuscript. The authors have thoroughly reviewed and edited all content and take full responsibility for the published work.

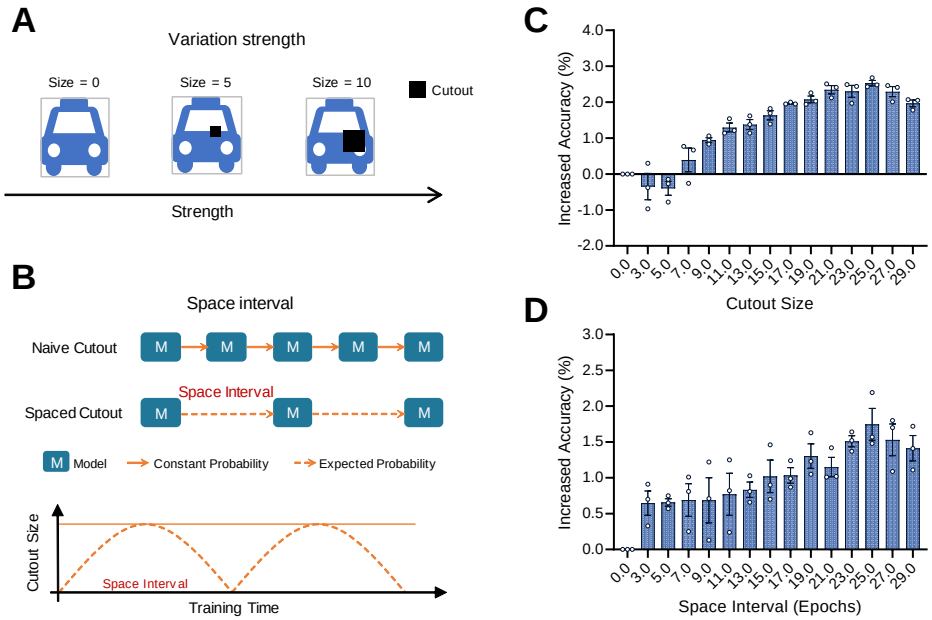

Figure S1: **Impact of variation strength and space interval in cutout augmentation.** (**A**) Variation strength is modulated by the cutout size, which controls the degree of occlusion applied to the input image. (**B**) Spaced interval determines how frequently the cutout size is altered across training epochs. (**C**) Test accuracy of ResNet-18 on CIFAR-10 exhibits a U-shaped trend with respect to cutout size. (**D**) Periodically varying cutout size with different spaced intervals also produces a U-shaped curve. All results are averaged over three runs with different random seeds. Data are presented as mean ± SEM.

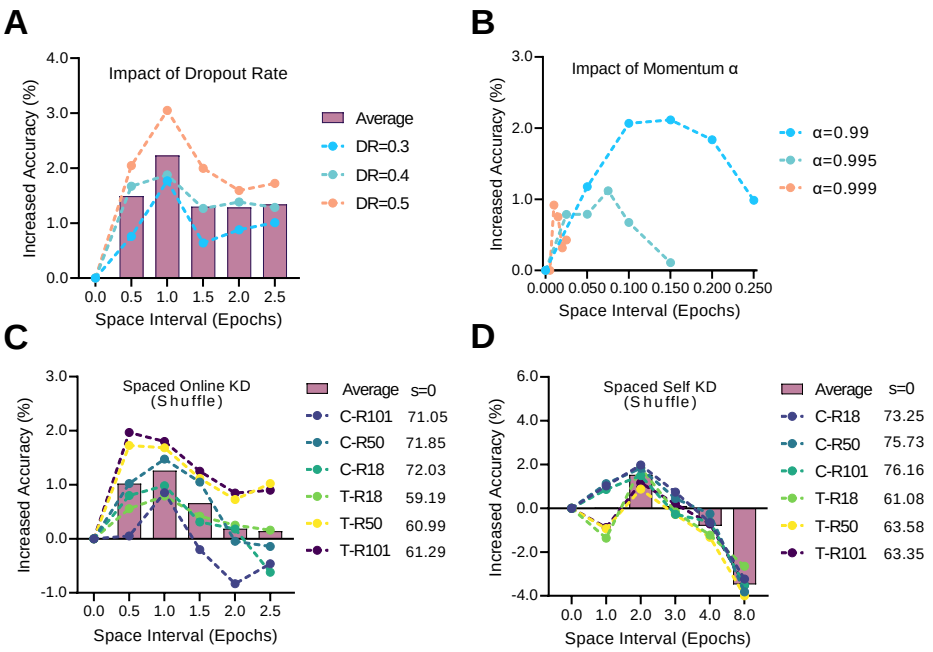

Figure S2: **Impact of variation strength and space interval on temporal ensemble strategies.** (**A**) Performance gains under different dropout rates ($p_{\max} \in \{0.3, 0.4, 0.5\}$) and spacing intervals using the spaced version of dropout. (**B**) Performance gains under different momentum coefficients ($\alpha \in \{0.99, 0.995, 0.999\}$) and spacing intervals using the spaced version of EMA. (**C**) Performance gains when teacher and student models receive shuffled training data at spaced intervals using spaced version of online KD. (**D**) Performance gains when teacher and student models receive shuffled training data at spaced intervals using spaced version of self KD. All results are averaged over three runs with different random seeds. Data are presented as mean ± SEM.

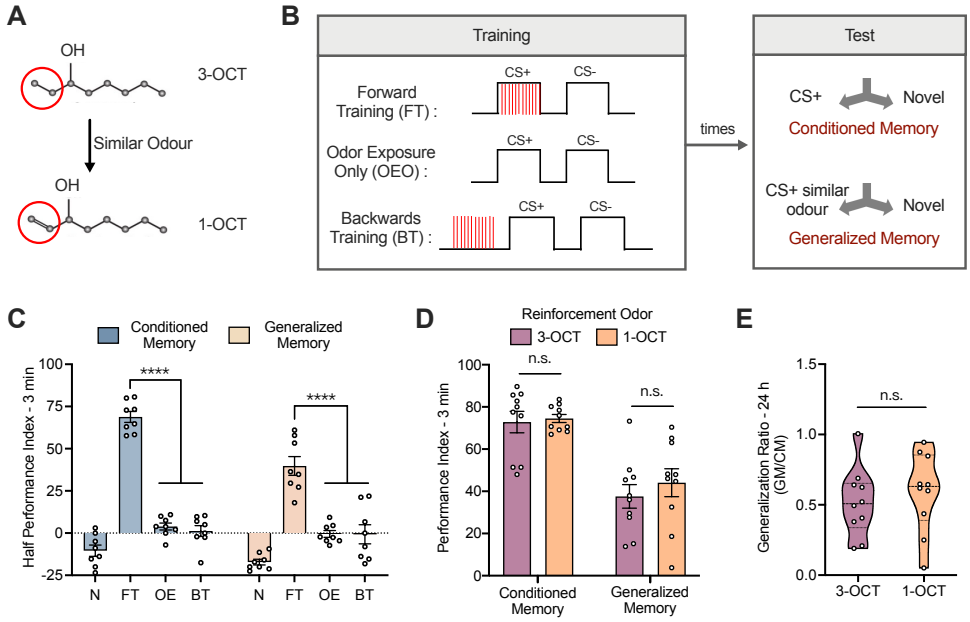

Figure S3: ***Drosophila* exhibit generalization to a similar odor after training.** (**A**) 3-OCT and 1-OCT are chemically similar odors. (**B**) Experimental paradigm for generalization training. Different groups of flies underwent forward training (FT), odor exposure only (OEO), or backwards training (BT). Conditioned and generalized memory are tested at indicated time points by presenting the CS+ odor (3-OCT) or a structurally similar odor (1-OCT), respectively, against a novel odor (MCH). (**C**) Both conditioned and generalized memory in the FT group are significantly higher than in control groups at 3 min. n=8. (**D**) Altering the reinforcement odor does not affect conditioned or generalized memory at 3 min. n=10. (**E**) Altering the reinforcement odor does not affect the generalization ratio at 3 min. n=10. Data are presented as mean ± SEM. Statistical significance is determined by two-way ANOVA with Sidak's multiple comparisons test or unpaired t test. ****p¡0.0001, n.s., not significant.

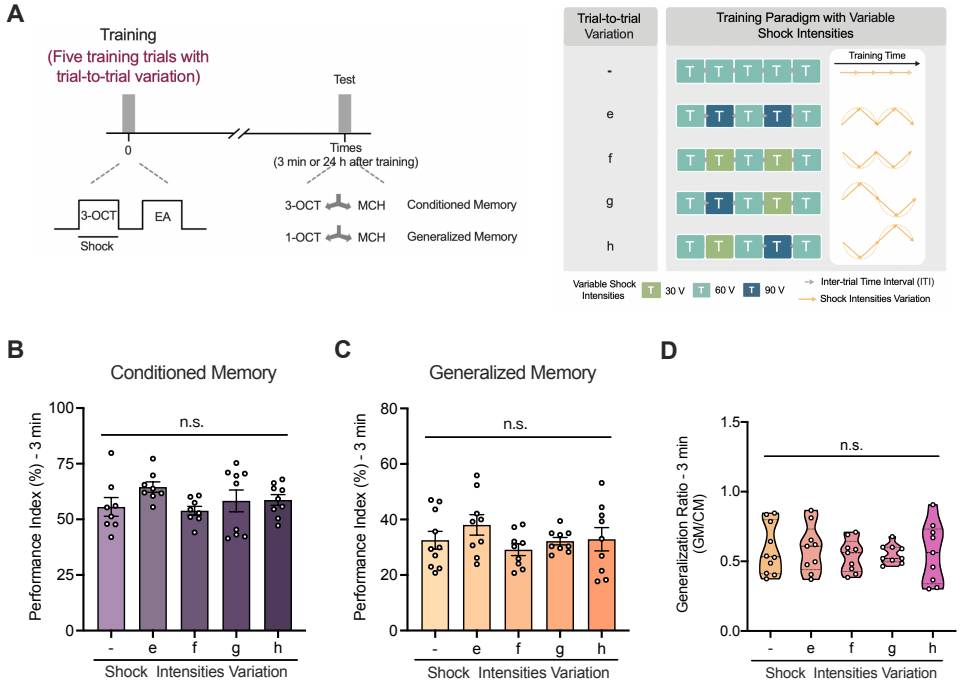

Figure S4: **Massed training with variable shock intensity has no effect on memory.** (**A**) Experimental paradigm for massed training with variable shock input. The control group (-) receives five trials with constant shock intensity, while experimental groups (e-h) receive trials with varying intensities to introduce encoding variability. (**B**-**D**) Conditioned memory (**B**, n=8-9), generalized memory (**C**, n=9-10), and generalization ratio (**D**, n=9-10) at 3 min are all no significant difference between control and experimental groups (e-h). Data are presented as mean ± SEM. Statistical significance is determined by one-way ANOVA with Dunnett's multiple comparisons test. n.s., not significant.

Algorithm 1 Spaced version of dropout

Require: model f_θ, dataset $\mathcal{D}_{\text{train}}$, max dropout rate $p_{\max}$, epoch number E, space interval s
Ensure: train f_θ with periodically varying dropout rates

```
for 1 ≤ e ≤ E do
    Compute dropout rate p_e ← p_max · sin(2πe/s)
    for (x_i, y_i) ∈ D_train do
        Sample dropout mask r ~ Bernoulli(p_e)
        Forward pass: h ← r ⊙ f_θ(x_i)
        Update θ ← θ − ∇_θ L_task(h, y_i)
    end for
end for
```

Algorithm 2 Spaced version of EMA

Require: model f_θ, dataset $\mathcal{D}_{\text{train}}$, momentum α, epoch number E, space interval s
Ensure: train f_θ with momentum updates every s steps

```
Initialize momentum model θ_m ← θ
Initialize step counter c ← 0
for 1 ≤ e ≤ E do
    for (x_i, y_i) ∈ D_train do
        Update θ ← θ − ∇_θ L_task(f_θ(x_i), y_i)
        c ← c + 1
        if c mod s = 0 then
            Update momentum model: θ_m ← αθ_m + (1 − α)θ
        end if
    end for
end for
```

Algorithm 3 Spaced version of online KD

Require: student f_θ, teacher g_ϕ, dataset $\mathcal{D}_{\text{train}}$, distillation loss coefficient λ, epoch number E, space interval s

Ensure: train student model using spaced online KD

```
Initialize data index set: R ← ∅
for 1 ≤ e ≤ E do
    for (x_i, y_i) ∈ D_train do
        R ← R ∪ {i}
        Update teacher φ ← φ − ∇_φ L_task(g_φ(x_i), y_i)
        if |R| == s then
            for j ∈ R do
                Retrieve (x_j, y_j) from D_train
                Update student θ ← θ − ∇_θ [λ L_KD(f_θ(x_j), g_φ(x_j)) + (1 − λ) L_task(f_θ(x_j), y_j)]
            end for
            Clear index set: R ← ∅
        end if
    end for
end for
```

Algorithm 4 Spaced version of self KD

Require: model $f_\theta = f_{\theta_1} \circ \cdots \circ f_{\theta_m}$ with m blocks, dataset $\mathcal{D}_{\text{train}}$, distillation loss coefficient λ, epoch number E, space interval s

Ensure: train f_θ using spaced self KD

```
Initialize data index set: R ← ∅
for 1 ≤ e ≤ E do
    for (x_i, y_i) ∈ D_train do
        R ← R ∪ {i}
        Update θ ← θ − ∇_θ L_task(f_θ(x_i), y_i)
        if |R| == s then
            for j ∈ R do
                Retrieve (x_j, y_j) from D_train
                Initialize L' ← L_task(f_θ(x_j), y_j)
                for 1 ≤ k < m do
                    L' ← L' + λ L_KD(f_θ1 ∘ ··· ∘ f_θk(x_j), f_θ(x_j))
                end for
                Update θ ← θ − ∇_θ L'
            end for
            Clear index set: R ← ∅
        end if
    end for
end for
```