EMERGENT SPECIALIZATION IN POPULATIONS OF SELF-SUPERVISED COLLABORATIVE VISION EXPERTS WITHOUT A SHARED GATE OR CROSS-AGENT GRADIENTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Consider a population of neural networks with no shared gate and no gradients flowing between them: each network has its own copy of the weights and is trained on its own, all see the same heterogeneous data, and any network may ask another for help through a forward pass. Does a division of labor emerge in such a population, and is it of any use? This is a candidate regime for training populations of models at scale, and it reverses how machine learning has organized a division of labor so far: in conventional approaches, such as mixtures of experts, one gate trained jointly with all experts decides who processes what, and specialization is typically assumed rather than measured. We answer the question in a controlled, small-scale proxy for predictive visual pretraining. Initially identical agents are fine-tuned with one shared objective, masked prediction of frozen DINOv3 features, on an unlabeled mixture of six visual domains, with no roles or central router. We measure *specialization* (does the best agent for an input align with the latent domain?) and *utilization* (is responsibility spread across the population?) along a ladder of regimes that removes central control step by step, ending in DISCO (DIStributed COllaboration), a fully local protocol in which each agent chooses a helper, reads its internal state through a gradient-free channel, and rewards its own router only for the local improvement the help produces. Specialization emerges, and it is useful: (i) a randomly routed population of the same size is worse than a single generalist while semantically routed populations are better, so specialization rather than population size is what helps; (ii) specialization survives the removal of the central router; (iii) the gradient-free exchange makes emergent expertise usable by non-experts: in DISCO alone a randomly chosen agent, helped by the expert, matches the solo generalist, while the experts themselves surpass it, and the effect persists on data outside the specialization mixture; (iv) local routers trained only on their own improvement select the emergent expert for 98% of inputs; (v) the effects persist across population size, agent capacity, data imbalance, and fine-tuning seeds. These results provide small-scale, measurable evidence for the population dynamics that decentralized predictive pretraining would require.

1 INTRODUCTION

Visual foundation models are mostly trained as single monolithic networks: one set of parameters processes every image, regardless of its domain, content, or source (Dosovitskiy et al., 2020; Oquab et al., 2024; Siméoni et al., 2025), and the same holds for video world models trained on increasingly heterogeneous data (Bruce et al., 2024; Agarwal et al., 2025; Assran et al., 2025). This works well, but leaves a basic question open: is a single network the right way to allocate capacity over heterogeneous data?

The alternative is a division of labor. Machine learning has explored it mainly through mixtures of experts (MoE) (Jacobs et al., 1991; Shazeer et al., 2017; Fedus et al., 2022), including sparse vision transformers (Riquelme et al., 2021; Puigcerver et al., 2024) and expert denoisers in diffusion models (Balaji et al., 2022; Feng et al., 2023; Fei et al., 2024). However, specialization in MoE is

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.

typically assumed rather than measured, and when it is measured, experts often organize around token position, noise level, or a small set of always-active experts rather than around semantic content (Fei et al., 2024; Wei et al., 2026; Wang et al., 2026; Huang et al., 2026). Moreover, MoE specialization is orchestrated: one gate, trained jointly with all experts in one computational graph, decides who processes what. This paper asks whether a division of labor also emerges among *separate* networks that route locally, without a shared gate and without gradients flowing between them, and whether it is of any use once it does, a question that has not been tested in a controlled way.

Why this setting matters. This is a candidate regime for training populations of models at scale. Without a shared gate, no global component lies on the path of every sample. Without cross-agent gradients, each agent is an independent optimization problem that can in principle run on separate hardware, be updated asynchronously, and be added to or removed from the population. Populations of separately trained models are already used to scale beyond one network, as branched and merged experts (Li et al., 2022; Gururangan et al., 2023; Sukhbaatar et al., 2024), decentralized paths (Douillard et al., 2024), or domain ensembles (Ersoy et al., 2025). All of this relies on an assumption that has not been tested in isolation: that a population of models, on its own, divides labor along the structure of its data, and that what one member learns can be used by the others. If it holds, decentralized populations become a candidate route to scaling on heterogeneous data; if not, central orchestration is necessary rather than optional.

A controlled proxy for predictive pretraining. Self-supervised pretraining is dominated by the prediction of missing or upcoming content: the next token in language models (Brown et al., 2020), the next frame in video world models (Bruce et al., 2024; Agarwal et al., 2025), and masked or future features in joint-embedding predictive architectures (Wei et al., 2022; Assran et al., 2023; Bardes et al., 2024) and latent world models (Zhou et al., 2025; Karypidis et al., 2025). Whether populations of such predictors develop the dynamics studied here should be verified at small scale first. We therefore use the simplest task that preserves the operation of predictive pretraining: agents reconstruct 70% masked patches in the frozen feature space of DINOv3 (Siméoni et al., 2025). The data are an unlabeled mixture of six visual domains outside the pretraining distribution, whose identity is the hidden factor that lets us measure specialization. Training a population takes hours, which makes a controlled study over regimes, population sizes, and capacities possible (Section 3.2).

Approach and questions. K agents start from the same pretrained weights and are fine-tuned with one shared self-supervised objective on the mixture, with no labels, roles, or central controller. We approach the target setting step by step along a ladder of training regimes with decreasing central control (Table 1) and ask: *(Q1)* Does a division of labor aligned with the latent domains improve on a single generalist, and is the improvement due to specialization rather than to having more models? *(Q2)* Does specialization still emerge when the central router is removed and every agent routes locally? *(Q3)* Can the expertise that emerges in one agent be used by another through a channel that carries no gradients? Three elements make the questions answerable. *(a) Metrics*: specialization is the normalized mutual information between the identity of the best agent for an input and the latent domain, and utilization the normalized entropy of that assignment; together they separate a division of labor from collapse and from load balancing alone. *(b) A testbed* with known latent structure in which all regimes share agents, initialization, and data budget, so that differences are attributable to routing and communication alone. *(c) The ladder*: a solo model; populations with random, privileged K-means, and learned central routing; *distributed delegation*, where each agent owns a router; and *DISCO* (DIStributed COllaboration), the target setting, where a requester agent selects a helper, reads the helper's internal state through a stop-gradient channel, and rewards its router only for the local improvement the help produces.

Findings. Specialization emerges in the target setting, and it is useful. A randomly routed population of the same size and budget is worse than the solo model while semantically routed populations are better, so specialization and not population size is what helps (Section 4.1). Specialization survives the removal of the central router (Section 4.2). The gradient-free exchange makes the emergent expertise usable: DISCO is the only regime in which a helped non-expert matches the solo model, it has the best worst-agent performance, and the effect persists on the ImageNet100 validation set, which the agents saw during pretraining but not during specialization (Section 4.3). Local routers trained

only on their own improvement select the emergent expert for 98% of inputs (Section 4.5), and the onset of specialization depends on population size and capacity in a predictable way (Section 4.4).

2 Related Work

Expert specialization in vision and generative models. Mixtures of experts route inputs to gated sub-networks (Jacobs et al., 1991; Shazeer et al., 2017; Fedus et al., 2022; Zhou et al., 2022); in vision they are used in ViTs (Riquelme et al., 2021; Puigcerver et al., 2024; Han et al., 2025), masked autoencoders (Liu et al., 2024), and diffusion transformers (Fei et al., 2024; Shi et al., 2025), and diffusion models have been split into expert denoisers by noise level (Balaji et al., 2022; Feng et al., 2023). In practice, gated specialization is fragile: representations collapse around expert centroids (Chi et al., 2022); diffusion experts specialize by position and timestep rather than content (Fei et al., 2024), motivating routing guidance (Wei et al., 2026), expert competition (Zheng et al., 2025), or spectral decomposition (Huang et al., 2026); and MoE language models exhibit a domain-invariant "standing committee" (Wang et al., 2026). All of these study experts inside one jointly optimized network with a shared gate. We study separate networks with local routers and no cross-network gradients, and measure specialization as the alignment of expert identity with latent structure rather than inferring it from routing statistics.

Populations of separately trained models. A separate body of work trains full models independently and composes them afterwards: branching and merging domain experts (Li et al., 2022; Sukhbaatar et al., 2024), discovering domains by clustering before training (Gururangan et al., 2023), heterogeneous domain ensembles (Ersoy et al., 2025), weight averaging and task arithmetic (Wortsman et al., 2022), routing among adapter libraries (Ostapenko et al., 2024; Muqeeth et al., 2024), and decentralized path composition (Douillard et al., 2024). In these systems the data partition is fixed in advance or expertise is composed only after training; our agents learn during training, from an unlabeled mixture, who handles what, and exchange information at inference time.

Communication between separate networks. Learned communication has been studied in multi-agent reinforcement learning (Sukhbaatar et al., 2016; Foerster et al., 2016; Jiang & Lu, 2018; Das et al., 2019), where roles can emerge among homogeneous agents (Wang et al., 2020; Mordatch & Abbeel, 2018; Baker et al., 2019), and in collaborative perception, where agents share intermediate features and learn when and with whom to communicate (Liu et al., 2020; Hu et al., 2022b; Wang et al., 2023). Closest in mechanism to ours are systems in which one network reads another's internal states through cross-attention, as in CALM (Bansal et al., 2024), and key–value communication between language models (Fu et al., 2026; Ye et al., 2025). These fix sender and receiver in advance and train a bridge for the pair. We ask whether a population can discover who should communicate with whom from the improvement a message brings, and whether keys and values are useful when the sender receives no gradient from the receiver; our agents also see the same input, so gains cannot come from fusing complementary views. No prior work isolates the combination studied here: identical agents, one unlabeled mixture, one shared self-supervised objective, local routing, and stop-gradient latent communication, with metrics that separate specialization from utilization.

3 A Controlled Testbed for Emergent Specialization

3.1 Measuring Specialization and Utilization

We assume that data are generated by a latent factor model: there is a latent variable $y \in \{1, \dots, M\}$ with prior $p(y)$ such that $p_{\mathrm{D}}(x) = \sum_y p(x \mid y)\, p(y)$. Consider a population of K agents $\mathcal{P}_K = \{a_i\}_{i=1}^K$, each evaluated on an input x through a loss $l(x, a_i)$. We call the agent with the lowest loss the *expert* on x, $e(x) = \arg\min_{1 \le i \le K} l(x, a_i)$, which assigns responsibility for each input to one agent. The population is *specialized* if this assignment correlates with y. We quantify the dependence with normalized mutual information, which is comparable across numbers of agents and factors,

$$\mathrm{Sp}(\mathcal{P}_K) = \mathrm{NMI}(e(x), y) = \frac{\mathrm{MI}(e(x), y)}{\min\{H(e(x)), H(y)\}} \in [0, 1]. \tag{1}$$

If one agent is the expert everywhere, $\mathrm{Sp} = 0$. However, Sp can be close to 1 when one agent handles almost all inputs and the others own tiny niches. The *utilization* $\mathrm{U}(\mathcal{P}_K) = H(e(x))/\ln K \in [0, 1]$, the normalized entropy of the expert assignment, separates a balanced division of labor from such collapse. A well-specialized population has high Sp *and* high U. Both are estimated on a validation set where y is known and depend only on the agents' standalone losses, not on any router, so a balanced router does not by itself produce high Sp: the distinction that routing statistics alone cannot make (Wang et al., 2026).

3.2 Experimental Setting, and Why We Chose It

Objective. Given an RGB image, a frozen DINOv3 encoder (Siméoni et al., 2025) produces patch features $z \in \mathbb{R}^{d\times h\times w}$. We sample a mask $m \in \{0, 1\}^{h\times w}$ that keeps a random rectangular window of 30% of the patches visible and hides the remaining 70%, and ask an agent to reconstruct the full feature map from $\hat{z} = m \cdot z$. Writing $x = (\hat{z}, z)$ for the resulting training input, the loss is the cosine distance on the masked positions, $l(x, a_i) = D_{\cos}(z, a_i(\hat{z}))$. All agents are small ViTs (Dosovitskiy et al., 2020) with 4–8 layers (Appendix A.1).

Performance measure. Cosine distance is a natural training loss but a poor performance measure, because outpainting is ambiguous. Instead of downstream probing (Zhou et al., 2025), we use a lightweight retrieval proxy: for each predicted feature at a hidden position we retrieve its five nearest neighbors (cosine) among the 196 ground-truth patch features of the same image and count a hit if the true position is among them; *R-Top5* is the hit rate over hidden positions, averaged over images (trivial baselines in Appendix A.10). For a population we report the expert envelope over the validation set, $\text{R-Top5-Best} = \frac{1}{|D_{\text{val}}|}\sum_{x\in D_{\text{val}}} \text{R-Top5}(x, a_{e(x)})$, the score of the expert (lowest-loss agent) on each input, and R-Top5-Worst, the score of the highest-loss agent. Best is the value the population would attain if every input reached its expert, a routing upper bound at fixed agents. It is selected by loss and scored by R-Top5, so it is not the metric's maximum over agents; Appendix A.10 reports how often the two coincide. Worst measures how much a non-expert loses. The performance actually attained lies in between and is reported at two operating points (Table 2): the *delegation* performance evaluates each sample with the agent that the regime's routing rule selects, and the *collaboration* performance lets a randomly chosen agent reconstruct it with the help of that selected agent, reading its keys and values. In the decentralized regimes the rule is the drawn agent's own router (Section 3.3).

Data. A single model is pretrained on ImageNet100 (Deng et al., 2009). All agents are then initialized from it and fine-tuned with LoRA (Hu et al., 2022a) on a mixture of six datasets that are out of distribution with respect to pretraining: AID (Xia et al., 2017) (aerial), KITTI (Geiger et al., 2013) (driving), BridgeDataV2 (Walke et al., 2023) and RT-1 (Brohan et al., 2022) (robot manipulation), CelebA (Liu et al., 2015) (faces), and 102 Flowers (Nilsback & Zisserman, 2008). By default we use 5,999 training and 2,190 validation images per dataset (35,994 and 13,140 in total; split construction and sequence handling in Appendix A.4). An imbalanced mixture D_{imb} with 4K, 4K, 8K, 9K, 7K, and 4K training images keeps the total of 36K fixed. The dataset index is the latent factor y used for evaluation and is never shown to any agent or router.

Why this setting. Each choice determines what the results can say. *One shared self-supervised objective.* Different losses or rewards would create roles by design; one loss on one unlabeled mixture is what makes a division of labor *emergent*. *Prediction in a frozen semantic space.* Like next-frame prediction and denoising, masked feature prediction infers missing content from context under a single loss. Working in DINOv3 space removes pixel-level ambiguity, lets a 4–8 layer ViT capture meaningful structure, and makes a population trainable in about six hours on four GPUs (Appendix A.3), so that regimes, population sizes, capacities, and mixtures can be swept. *A known latent factor.* Measuring specialization needs ground truth: six visually distinct domains outside the pretraining distribution give a factor y that is well defined, unseen during pretraining, and hidden during fine-tuning. *Whole-sample routing among separate networks* keeps every agent a standalone learner and makes specialization interpretable at the level of the data.

Table 1: The ladder of training regimes. Each step removes one form of central control. All population regimes share the same K agents, backbones, initialization, LoRA budget, and total training samples.

Regime	Router	Router signal	Training	Role in the study
Solo	–	–	–	Generalist reference
Random	central, fixed	none (random split)	non-local	Control: population without semantics
K-means	central, fixed	clusters of clean DINOv2 features	non-local	Positive control: privileged split
Learned central	central, trained	policy gradient on helper loss	non-local	MoE-style reference: one central gate trained online
Distr. delegation	one per agent	policy gradient on helper loss	non-local	Q2: no central router
DISCO (ours)	one per agent	policy gradient on *improvement*	local	Q3: is expertise shareable?

3.3 A Ladder of Training Regimes: From Central Control to Local Collaboration

The regimes in Table 1 lead from full central control to the target setting. The first four are diagnostic controls built from standard components. Distributed delegation removes the central router, but a sample still has to be passed to the agent that solves it best. DISCO additionally lets the requester reconstruct the sample itself *with* the selected agent's help. In every population regime a *router* $r\colon x \mapsto \{1, \dots, K\}$ maps its input to the index of an agent, and each agent is trained on the samples routed to it. The input to the router is the corrupted sample $\hat{z}$ in every learned regime, and the clean features only for the K-means control. The reconstruction loss of a batch $\{x_i\}_{i=1}^{b}$ with routing r is $\mathcal{L}_{\text{rec}} = \frac{1}{b}\sum_{i=1}^{b} l(x_i, a_{r(x_i)})$, so that routing changes the effective data distribution seen by each agent and is the only mechanism through which specialization can arise.

Solo, random, and K-means routing. The *solo* model is one agent trained on the full mixture, with no routing. *Random* routing splits the training set into K random fixed subsets and trains each agent on its own subset: the population has K agents and the same data budget but no semantic assignment. It isolates the effect of having several models from the effect of specialization. *K-means* routing clusters the training set offline into K groups, using the `[CLS]` features that a frozen DINOv2 ViT-B/14 encoder (Oquab et al., 2024) computes on the *uncorrupted* images, and assigns each group to an agent. It is a positive control with privileged information.

Learned central routing. A trainable router r_θ (a two-layer MLP on the mean of the *visible* DINO tokens; it sees only what an agent sees) outputs a distribution over agents. For sample i the assigned agent j is sampled, $j \sim r_\theta(\hat{z}_i)$. Since sampling is not differentiable, the router is trained with a policy-gradient loss added to the reconstruction loss,

$$\mathcal{L}_{\text{router}} = -\frac{1}{b}\sum_{i=1}^{b} \left(R_i - \bar{R}\right) \log r_\theta^{(j)}(\hat{z}_i), \qquad R_i = -\text{sg}\left[l(x_i, a_j)\right], \tag{2}$$

where $\bar{R}$ is an exponential moving average of the rewards and $\text{sg}[\cdot]$ denotes the stop-gradient operator. This is the closest analogue of an MoE gate in our setting: one online router, trained jointly with the agents, but routing whole samples between separate networks rather than tokens between sub-modules (Jacobs et al., 1991; Fedus et al., 2022). All learned routers in this paper additionally receive a small KL penalty toward a uniform marginal routing distribution (Appendix A.6), the standard load-balancing mechanism of MoEs; Section 4.6 reports what happens without it.

Distributed delegation: removing the central router. We now give every agent its own routing head. For each training batch a *requester* a_q is drawn uniformly at random. Its router assigns each sample i in the batch to a *helper* $j \sim r_q(\hat{z}_i)$, which is trained on the sample as before, and r_q is trained with Eq. (2). Each head is a two-layer MLP on the mean of the requester's own queries, so a router reuses its agent's features but has its own parameters. The requester is excluded from its own routing distribution (Section 4.6 lifts this exclusion). No central component is left, but the learning signal is still not local: a router is rewarded by *another* agent's loss, and the requester only finds an

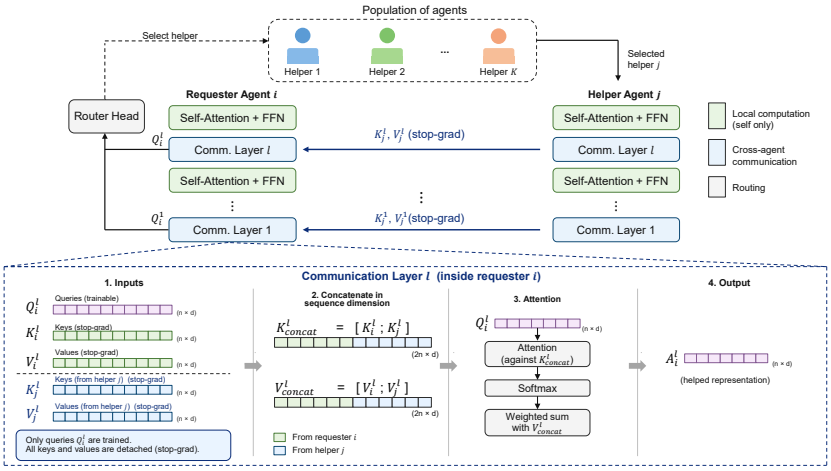


Figure 1: DISCO. A random requester routes each sample to a helper with its own router; in communication layers its queries attend to its own and the helper's detached keys and values. The helper is trained on the sample, the requester on its helped reconstruction, the router on the improvement.

agent that solves the sample better, so the population can never do better than its agents do alone on their niches.

DISCO: distributed collaboration. We want the requester to *use* the helper rather than hand the sample over, so that expertise can flow across the population. DISCO therefore changes two things with respect to delegation, the channel and the router reward, and Section 4.6 separates their effects. Training the requester on every sample as well would restore the symmetry between agents and remove the specialization. Instead, the helper communicates part of its internal computation and the requester learns to reconstruct *with* that message. Two design decisions define the protocol.

(1) Messages are keys and values with stopped gradients. A differentiable channel that backpropagates the requester's loss into the helper would train helpers to produce useful messages, but it would also couple all agents into one optimization graph, the MoE regime we depart from. We therefore stop all gradients through the channel and build messages from quantities the helper already computes for itself. The l-th self-attention block of agent a_i outputs $A_{\text{solo}}(a_i, l) = \text{SoftMax}(Q_i^l K_i^{l\top}/\sqrt{d})\, V_i^l$ with $Q_i^l, K_i^l, V_i^l \in \mathbb{R}^{n\times d}$. In a *communication layer*, after requester a_i has selected helper a_j, the helper's keys and values are concatenated to the requester's own:

$$A_{\text{helped}}(a_i, a_j, l) = \text{SoftMax}\left(\frac{Q_i^l K_{\text{cat}}^{l\top}}{\sqrt{d}}\right) V_{\text{cat}}^l, \quad K_{\text{cat}}^l = \text{sg}\left[K_i^l;\, K_j^l\right],\; V_{\text{cat}}^l = \text{sg}\left[V_i^l;\, V_j^l\right], \quad (3)$$

where $[\cdot;\cdot]$ concatenates along the token axis. The requester is free to ignore the helper's entries, and communication can be switched on per layer (Table 4). A requester thus has a solo loss $l_{\text{solo}}(x, a_i)$ and a helped loss $l_{\text{helped}}(x, a_i, a_j)$. Figure 1 illustrates the pipeline.

(2) Routers are rewarded by improvement, not by the helper's accuracy. In delegation the router learns who is best in isolation. In DISCO the router of requester a_q is trained with Eq. (2) using $R_i = \text{sg}[\, l_{\text{solo}}(x_i, a_q) - l_{\text{helped}}(x_i, a_q, a_j)\,]$, the reduction in the requester's own loss due to the help. This signal is local to the requester and never uses the helper's standalone performance. Section 4.5 asks whether such routers nonetheless find the experts. The full objective is

$$\mathcal{L}_{\text{DISCO}} = \lambda_{\text{rec}}\mathcal{L}_{\text{rec}} + \lambda_{\text{router}}\mathcal{L}_{\text{router}} + \lambda_{\text{helped}}\mathcal{L}_{\text{helped}}, \qquad \mathcal{L}_{\text{helped}} = \frac{1}{b}\sum_{i=1}^{b} l_{\text{helped}}(x_i, a_q, a_j), \quad (4)$$

where $\mathcal{L}_{\text{rec}}$ trains the helpers on their assigned samples as in all other regimes. Algorithm 1 in Appendix A.6 summarizes one step. Every component is standard (ViTs, LoRA, attention, REINFORCE); what is specific to DISCO is their combination, local improvement-based routing with stop-gradient key–value communication.

Table 2: Routing study. Best/Worst is the expert envelope (R-Top5 of the lowest-/highest-loss agent). Deleg./Collab.: performance with the regime's own routing rule, without and with communication, for a random requester. Worst+help: the worst agent on an input reconstructing with the selected agent. Means over three draws of crops and masks shared by all regimes; 95% paired bootstrap intervals within ± 0.003 per entry and ± 0.0015 for differences between regimes (Appendix A.10).

Regime	Sp↑	U↑	Best↑	Worst↑	Deleg.	Collab.	Worst+help
Solo	–	–	0.653	0.653	–	–	–
Random	0.011	0.999	0.650	0.634	0.642	0.643	0.637
K-means	**0.957**	**0.962**	**0.678**	0.535	**0.678**	0.618	0.582
Learned central	0.788	**0.995**	**0.678**	0.533	**0.678**	0.620	0.592
Distr. delegation	0.926	0.846	**0.672**	**0.546**	**0.665**	**0.626**	**0.593**
DISCO	**0.931**	0.841	0.668	**0.568**	0.662	**0.653**	**0.644**

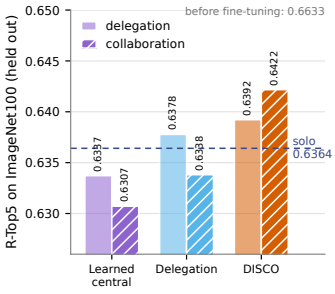


Figure 2: Retention on the ImageNet100 validation set (5,000 images, seen during pretraining, unseen during specialization; protocol of Table 2). All populations lose accuracy with respect to the model before fine-tuning. DISCO with communication loses least and is the only regime in which communication helps.

Table 3: Same requester, same input, different message. R-Top5 of a requester reconstructing with the message in the row, averaged over the four requesters ("all"), or over the requesters that are not the expert on the input ("non-exp."). Selected helper is the Collab. column of Table 2. Three draws; 95% paired intervals on differences between rows are within ± 0.002.

	DISCO		Delegation	
Message	all	non-exp.	all	non-exp.
alone (no message)	0.618	0.601	0.602	0.578
selected helper	0.653	0.649	0.626	0.616
expert (lowest loss)	0.654	0.649	0.630	0.616
random other agent	0.636	0.628	0.609	0.597
own K/V duplicated	0.618	0.601	0.602	0.578
selected, other image's K/V	0.380	0.366	0.404	0.403
all-zero K/V	0.608	0.590	0.594	0.570

4 Results

In this section, we follow the ladder of Table 1, one question per subsection. Unless stated otherwise, populations have $K = 4$ type-B agents, i.e., base size with 43M parameters each (Table 4), are trained on the balanced mixture, and all regimes see the same number of training samples. All numbers are means over three draws of crops and masks on the validation set, and the intervals we quote are paired bootstrap intervals over images (Appendix A.10). They measure evaluation-set uncertainty for the trained checkpoints, not variability across training runs, which Section 4.6 addresses separately.

4.1 Q1: Does Specialization Matter, and Is It Specialization or Just More Models?

As Table 2 shows (plotted in Appendix A.7), central routing, privileged or learned, produces specialized, well-utilized populations ($\mathrm{Sp} = 0.79$–0.96, $\mathrm{U} = 0.96$–1.00) and raises R-Top5-Best from 0.653 (solo) to 0.678. The K-means control specializes most, while the learned router, which sees only the masked input, recovers the full gain online with a less semantic split.

The random-routing control is what gives these numbers their meaning. It uses the same four agents, backbones, and data budget, and its best-agent responsibility is balanced ($\mathrm{U} = 1.00$), but it carries no semantic structure ($\mathrm{Sp} = 0.01$) and performs *worse* than the solo model, since each agent sees a quarter of the data: several models are not enough, and balanced load is not specialization. The gain

appears only when responsibility aligns with latent structure. At the same time the Worst endpoint drops from 0.653 to ≈ 0.53: a specialized agent is a poor generalist, which is why sharing expertise (Q3) matters.

4.2 Q2: Does Specialization Survive the Removal of the Central Router?

Distributed delegation replaces the single gate by one router per agent, each trained only on the batches for which its agent is the requester. Specialization nevertheless emerges ($\mathrm{Sp} = 0.93$), close to the K-means split and above the learned central router (0.79), and R-Top5-Best (0.672) is within 0.006 of the central regimes. Utilization is lower ($\mathrm{U} = 0.85$): without a global view the population balances responsibility less evenly. Nothing in the local routers' objective refers to domains, yet the population is almost as domain-aligned as an offline clustering of clean features, grouping the six domains into three two-domain niches with one agent almost idle (Appendix A.9).

4.3 Q3: Can Emergent Expertise Be Shared Through a Stop-Gradient Channel?

DISCO specializes as much as delegation ($\mathrm{Sp} = 0.93$, $\mathrm{U} = 0.84$; niches in Appendix A.9) at a slightly lower expert Best (0.668 vs. 0.672). First, DISCO has the highest R-Top5-Worst of the specialized populations (0.568 vs. 0.533–0.546). Second, communication recovers a non-expert to generalist level: a random requester reconstructing with its selected helper reaches 0.653, on par with the solo model and 0.027 above the same measurement in delegation (0.626). The same requester alone scores 0.618, so the message is worth 0.035 to it. Collaboration does not beat delegating to the expert on the mixture (0.653 vs. 0.662); its value is that a non-expert becomes as good as a generalist while the experts remain better than one. The worst agent on an input, once helped, improves from 0.568 to 0.644 (delegation: 0.546 to 0.593; Figure 7 shows what the help restores). Third, on the ImageNet100 validation set, which the agents saw during pretraining but not during specialization (a retention test rather than transfer to a new distribution), every population and the solo model fall below the model before fine-tuning (0.663). DISCO with communication loses least (0.642 vs. 0.636 solo, 0.634 delegation, 0.631 learned central; paired difference to solo $+0.006$, interval $[0.005, 0.006]$), and it is the only regime in which communication raises performance above the regime's own delegation (Figure 2). All differences in this section are resolved by the paired intervals (Appendix A.10).

4.4 Q4: What Governs When Specialization Emerges?

Population size. We train DISCO with 4, 6, and 8 agents (Figure 5a,b, Appendix A.7). All populations reach high specialization and larger ones end with higher R-Top5-Best but specialize *later*: requesters are drawn uniformly, so each router is trained on fewer batches and the symmetry takes longer to break.

Agent capacity. With S, B, and L agents (14M, 43M, 103M parameters; Table 4) in populations of four, performance grows with capacity, and more capable agents specialize *earlier* (Figure 5c,d). Larger agents produce more informative queries, keys, and values, so the improvement signal tells helpers apart sooner.

Data imbalance. On the imbalanced mixture D_{imb}, specialization is somewhat weaker ($\mathrm{Sp} = 0.75$), utilization higher ($\mathrm{U} = 0.93$), performance slightly lower, and the grouping of domains changes (Appendix A.9); the population does not collapse onto the dominant domains.

4.5 Q5: Do Local Routers Find the Experts?

The population *agrees* on an input if the most probable helper $h_i(\hat{z}) = \arg\max_j r_i^{(j)}(\hat{z})$ of every non-expert requester $i \neq e(x)$ is the expert. The 4-agent DISCO population agrees on 98.3% of the validation inputs (delegation: 97.9%; 6 agents 98%, 8 agents 88%, Table 8). The routers never saw a helper's standalone loss; that they converge on the loss-defined expert means that "most helpful to me" and "best at this input" coincide, so the division of labor is visible to the agents themselves (attention patterns in Appendix A.9).

4.6 Q6: Ablations and Robustness

Same requester, different messages. Table 3 evaluates every requester on every validation input while only the message changes. A DISCO requester scores 0.618 alone, 0.653 with the helper its router selects, and 0.636 with a random other agent's message. Information-free messages of the same shape isolate the content: duplicating the requester's own keys and values changes nothing, all-zero keys and values cost 0.010, and the selected helper's keys and values computed for a *different* image cost 0.239: the requester reads the message. Non-expert requesters gain 0.048 while the expert loses 0.004. Delegation agents, never trained to read a message, show the same ordering with smaller values.

Training seeds at a quarter of the budget. Appendix A.8 repeats the two distributed regimes with three new seeds for 100 fine-tuning epochs. Specialization emerges in every run ($\mathrm{Sp} = 0.92$–0.95 for DISCO, 0.86–0.95 for delegation), the ladder of Section 4.1 replicates, and in every seed DISCO trades an expert Best within 0.002 of delegation for a higher Worst (0.594 vs. 0.568), collaboration (0.631 vs. 0.622), and Worst+help (0.625 vs. 0.607). What varies is the shape of the division of labor: utilization ranges from 0.85 to 0.96 in both regimes, and whether DISCO specializes *more* than delegation is seed-dependent (0.94 ± 0.02 vs. 0.89 ± 0.05) while its advantage in Worst and in the helped scores is not.

Reward versus channel. Two runs at the same budget separate the two changes DISCO makes to delegation (Table 9). Training the requester on its helped reconstruction while keeping delegation's router reward reproduces the helped scores of DISCO (collaboration 0.631, Worst+help 0.624, Worst 0.586) with delegation's specialization ($\mathrm{Sp} = 0.86$) and routing agreement (82% vs. 98%): learning to read the message, not the improvement reward, creates the gain of a helped requester, while the improvement reward is what aligns the routers with the expert. Communication training under random routing yields neither specialization ($\mathrm{Sp} = 0.00$) nor a benefit from messages (0.627 helped vs. 0.626 alone): messages are worth reading only once routing has made the agents different.

Protocol choices. Without the KL term toward a uniform routing marginal ($\lambda_{\mathrm{KL}} = 0$), the local routers of DISCO collapse within four epochs onto one agent, which becomes a generalist expert on 90% of the inputs ($\mathrm{Sp} = 0.08$, $\mathrm{U} = 0.25$, Best 0.633, the solo level): the load-balancing pressure that MoEs also use is a precondition for the division of labor, while the random-routing control shows that balance alone does not produce one; what emerges is the alignment of the balanced niches with the domains. Allowing a requester to select itself slows and weakens specialization ($\mathrm{Sp} = 0.81$ after 72 epochs, all four agents used) but does not prevent it: a router selects its own agent for 91% of the inputs on which that agent is the expert and for only 2% of the others, and the helped scores equal those of DISCO.

5 Discussion

We studied emergent specialization in populations of initially identical neural agents trained with a shared self-supervised reconstruction objective, without task labels, predefined roles, or a central orchestrator. Using specialization and utilization as metrics, we showed that agents organize around the latent structure of heterogeneous visual data as soon as routing is tied to per-sample performance, and that this improves on both a single agent and a non-specialized population of the same size and budget. What a router is trained on, rather than how evenly it spreads the load, makes the division of labor semantic, and simple local routing with constrained communication suffices to support it.

We introduced DISCO, a distributed collaborative routing framework in which agents locally select helpers and communicate through attention keys and values with stopped gradients. Across our experiments DISCO induces strong specialization, is the only regime of Table 2 in which a requester gains from the message it receives, and remains effective across population size, agent capacity, and data balance. Future work involves larger, more heterogeneous populations, richer messages, and other predictive objectives.

AI USE STATEMENT

Among the tasks for which ICLR requires disclosure, we used generative AI tools to provide feedback on experiments, implement methods, and interpret results: AI assistance wrote and edited the code that ran baselines and collected the evaluation metrics, commented on experimental design, and proposed readings of the measurements. We did not use generative AI to develop the conceptual framework, to propose or refine hypotheses, to clean or reformat datasets, or for qualitative and thematic data analysis; the research question, the testbed, the ladder of routing regimes, the specialization and utilization metrics, and the DISCO protocol are the authors' own. Generating synthetic data, formulating or proving mathematical claims, and translation do not apply: all data are public real-image datasets and the paper has no theoretical results.

Among the tasks for which disclosure is recommended, we used generative AI to draft, restructure, and edit parts of the paper, propose alternative titles, create figures, tables, and software code, search for and summarize related work, and assess the novelty of our idea against that literature.

We reviewed all AI-assisted work: every AI-suggested reference was checked against the original publication, the novelty assessment only directed us to papers we then evaluated ourselves, all AI-assisted code was reviewed by the authors, and AI-suggested interpretations were kept only where the measurements supported them. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

All architectures, hyper-parameters, loss weights, routing temperatures, regularizers, and the evaluation protocol are specified in Section 3 and Appendix A, including the router architecture, optimizer settings, exact trainable-parameter counts, the construction of every data split (Appendix A.4), and the per-sample compute of every regime (Appendix A.5); the datasets are public and the exact data budget per domain is stated in Section 3.2. The main runs share one seed and data order; Section 4.6 reports independent fine-tuning seeds. Code and configuration files for every regime will be released upon acceptance.

ETHICS STATEMENT

This is foundational research on modular self-supervised vision models trained on public datasets; we foresee no direct societal harm from the experiments. A broader discussion of potential impacts, including the risk of uneven performance across domains or groups when specialization aligns with sensitive attributes, is given in Appendix C.

REFERENCES

Niket Agarwal, Arslan Ali, Maciej Bala, Yogesh Balaji, Erik Barker, Tiffany Cai, et al. Cosmos world foundation model platform for physical AI. *arXiv preprint arXiv:2501.03575*, 2025.

Mahmoud Assran, Quentin Duval, Ishan Misra, Piotr Bojanowski, Pascal Vincent, Michael Rabbat, Yann LeCun, and Nicolas Ballas. Self-supervised learning from images with a joint-embedding predictive architecture. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

Mido Assran, Adrien Bardes, David Fan, Quentin Garrido, Russell Howes, Mojtaba Komeili, Matthew Muckley, Ammar Rizvi, et al. V-jepa 2: Self-supervised video models enable understanding, prediction and planning. *arXiv preprint arXiv:2506.09985*, 2025.

Bowen Baker, Ingmar Kanitscheider, Todor Markov, Yi Wu, Glenn Powell, Bob McGrew, and Igor Mordatch. Emergent tool use from multi-agent autocurricula. *arXiv preprint arXiv:1909.07528*, 2019.

Yogesh Balaji, Seungjun Nah, Xun Huang, Arash Vahdat, Jiaming Song, Qinsheng Zhang, Karsten Kreis, Miika Aittala, Timo Aila, Samuli Laine, Bryan Catanzaro, Tero Karras, and Ming-Yu Liu. eDiff-I: Text-to-image diffusion models with an ensemble of expert denoisers. *arXiv preprint arXiv:2211.01324*, 2022.

Rachit Bansal, Bidisha Samanta, Siddharth Dalmia, Nitish Gupta, Sriram Ganapathy, Abhishek Bapna, Prateek Jain, and Partha Talukdar. Llm augmented llms: Expanding capabilities through composition. In *International Conference on Learning Representations*, volume 2024, pp. 71–87, 2024.

Adrien Bardes, Quentin Garrido, Jean Ponce, Xinlei Chen, Michael Rabbat, Yann LeCun, Mido Assran, and Nicolas Ballas. Revisiting feature prediction for learning visual representations from video. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856. URL https://openreview.net/forum?id=QaCCuDfBk2. Featured Certification.

Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Joseph Dabis, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, Jasmine Hsu, Julian Ibarz, Brian Ichter, Alex Irpan, Tomas Jackson, Sally Jesmonth, Nikhil Joshi, Ryan Julian, Dmitry Kalashnikov, Yuheng Kuang, Isabel Leal, Kuang-Huei Lee, Sergey Levine, Yao Lu, Utsav Malla, Deeksha Manjunath, Igor Mordatch, Ofir Nachum, Carolina Parada, Jodilyn Peralta, Emily Perez, Karl Pertsch, Jornell Quiambao, Kanishka Rao, Michael Ryoo, Grecia Salazar, Pannag Sanketi, Kevin Sayed, Jaspiar Singh, Sumedh Sontakke, Austin Stone, Clayton Tan, Huong Tran, Vincent Vanhoucke, Steve Vega, Quan Vuong, Fei Xia, Ted Xiao, Peng Xu, Sichun Xu, Tianhe Yu, and Brianna Zitkovich. Rt-1: Robotics transformer for real-world control at scale. In *arXiv preprint arXiv:2212.06817*, 2022.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

Jake Bruce, Michael Dennis, Ashley Edwards, Jack Parker-Holder, Yuge Shi, Edward Hughes, Matthew Lai, Aditi Mavalankar, Richie Steigerwald, Chris Apps, et al. Genie: Generative interactive environments. In *International Conference on Machine Learning (ICML)*, 2024.

Zewen Chi, Li Dong, Shaohan Huang, Damai Dai, Shuming Ma, Barun Patra, Saksham Singhal, Payal Bajaj, Xia Song, Xian-Ling Mao, Heyan Huang, and Furu Wei. On the representation collapse of sparse mixture of experts. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

Abhishek Das, Théophile Gervet, Joshua Romoff, Dhruv Batra, Devi Parikh, Mike Rabbat, and Joelle Pineau. TarMAC: Targeted multi-agent communication. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 1538–1546. PMLR, 2019. URL https://proceedings.mlr.press/v97/das19a.html.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255. Ieee, 2009.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

Arthur Douillard, Qixuan Feng, Andrei A Rusu, Adhiguna Kuncoro, Yani Donchev, Rachita Chhaparia, Ionel Gog, Marc'Aurelio Ranzato, Jiajun Shen, and Arthur Szlam. Dipaco: Distributed path composition. *arXiv preprint arXiv:2403.10616*, 2024.

Oğuzhan Ersoy, Jari Kolehmainen, and Gabriel Passamani Andrade. HDEE: Heterogeneous domain expert ensemble. *arXiv preprint arXiv:2502.19385*, 2025.

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. URL https://www.jmlr.org/papers/v23/21-0998.html.

Zhengcong Fei, Mingyuan Fan, Changqian Yu, Debang Li, and Junshi Huang. Scaling diffusion transformers to 16 billion parameters. *arXiv preprint arXiv:2407.11633*, 2024.

Zhida Feng, Zhenyu Zhang, Xintong Yu, Yewei Fang, Lanxin Li, Xuyi Chen, Yuxiang Lu, Jiaxiang Liu, Weichong Yin, Shikun Feng, Yu Sun, Li Chen, Hao Tian, Hua Wu, and Haifeng Wang. ERNIE-ViLG 2.0: Improving text-to-image diffusion model with knowledge-enhanced mixture-of-denoising-experts. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

Jakob N. Foerster, Yannis M. Assael, Nando de Freitas, and Shimon Whiteson. Learning to communicate with deep multi-agent reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 29, pp. 2137–2145, 2016. URL https://papers.nips.cc/paper/6042-learning-to-communicate-with-deep-multi-agent-reinforcement-learning.

Tianyu Fu, Zihan Min, Hanling Zhang, Jichao Yan, Guohao Dai, Wanli Ouyang, and Yu Wang. Cache-to-cache: Direct semantic communication between large language models. In *International Conference on Learning Representations*, volume 2026, pp. 43130–43158, 2026.

Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *The international journal of robotics research*, 32(11):1231–1237, 2013.

Suchin Gururangan, Margaret Li, Mike Lewis, Weijia Shi, Tim Althoff, Noah A. Smith, and Luke Zettlemoyer. Scaling expert language models with unsupervised domain discovery. *arXiv preprint arXiv:2303.14177*, 2023.

Xumeng Han, Longhui Wei, Zhiyang Dou, Yingfei Sun, Zhenjun Han, and Qi Tian. Vimoe: An empirical study of designing vision mixture-of-experts. *IEEE Transactions on Image Processing*, 34:7209–7221, 2025.

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*, 2022a. URL https://openreview.net/forum?id=nZeVKeeFYf9.

Yue Hu, Shaoheng Fang, Zixing Lei, Yiqi Zhong, and Siheng Chen. Where2comm: Communication-efficient collaborative perception via spatial confidence maps. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022b.

Ruijun Huang, Fang Dong, Xin Zhang, Anrui Chen, Hengjie Cao, Zhendong Huang, Jixian Zhou, Mengyi Chen, Yifeng Yang, Mingzhi Dong, Yujiang Wang, Jinlong Hou, Qin Lv, Robert P. Dick, Yuan Cheng, Fan Yang, Tun Lu, Chun Zhang, and Li Shang. SD-moe: Spectral decomposition for effective expert specialization. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=FnB7lFzsJz.

Robert A. Jacobs, Michael I. Jordan, Steven J. Nowlan, and Geoffrey E. Hinton. Adaptive mixtures of local experts. *Neural Computation*, 3(1):79–87, 1991. doi: 10.1162/neco.1991.3.1.79. URL https://direct.mit.edu/neco/article/3/1/79/5560/Adaptive-Mixtures-of-Local-Experts.

Jiechuan Jiang and Zongqing Lu. Learning attentional communication for multi-agent cooperation. In *Advances in Neural Information Processing Systems*, volume 31, pp. 7265–7275, 2018. URL https://papers.nips.cc/paper/7956-learning-attentional-communication-for-multi-agent-cooperation.

Efstathios Karypidis, Ioannis Kakogeorgiou, Spyridon Gidaris, and Nikos Komodakis. Dino-foresight: Looking into the future with dino. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pp. 163779–163811. Curran Associates, Inc., 2025. doi: 10.52202/085713-5466. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/efaca9631eb6bd5df8f6761c5d9a9fe3-Paper-Conference.pdf.

Margaret Li, Suchin Gururangan, Tim Dettmers, Mike Lewis, Tim Althoff, Noah A. Smith, and Luke Zettlemoyer. Branch-train-merge: Embarrassingly parallel training of expert language models. *arXiv preprint arXiv:2208.03306*, 2022.

Yen-Cheng Liu, Junjiao Tian, Nathaniel Glaser, and Zsolt Kira. When2com: Multi-agent perception via communication graph grouping. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

Zhili Liu, Kai Chen, Jianhua Han, Lanqing Hong, Hang Xu, Zhenguo Li, and James T. Kwok. Task-customized masked autoencoder via mixture of cluster-conditional experts. *arXiv preprint arXiv:2402.05382*, 2024.

Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, December 2015.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

Igor Mordatch and Pieter Abbeel. Emergence of grounded compositional language in multi-agent populations. In *AAAI Conference on Artificial Intelligence*, 2018.

Mohammed Muqeeth, Haokun Liu, Yufan Liu, and Colin Raffel. Learning to route among specialized experts for zero-shot generalization. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org, 2024.

Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *Indian Conference on Computer Vision, Graphics and Image Processing*, Dec 2008.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Mahmoud Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Hervé Jégou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. DINOv2: Learning robust visual features without supervision. *Transactions on Machine Learning Research*, 2024. URL https://openreview.net/forum?id=a68SUt6zFt.

Oleksiy Ostapenko, Zhan Su, Edoardo Ponti, Laurent Charlin, Nicolas Le Roux, Lucas Caccia, and Alessandro Sordoni. Towards modular LLMs by building and reusing a library of loRAs. In *Forty-first International Conference on Machine Learning*, 2024. URL https://openreview.net/forum?id=0ZFWfeVsaD.

Joan Puigcerver, Carlos Riquelme, Basil Mustafa, and Neil Houlsby. From sparse to soft mixtures of experts. In *International Conference on Learning Representations (ICLR)*, 2024.

Carlos Riquelme, Joan Puigcerver, Basil Mustafa, Maxim Neumann, Rodolphe Jenatton, André Susano Pinto, Daniel Keysers, and Neil Houlsby. Scaling vision with sparse mixture of experts. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

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. In *International Conference on Learning Representations*, 2017. URL https://openreview.net/forum?id=B1ckMDqlg.

Minglei Shi, Ziyang Yuan, Haotian Yang, Xintao Wang, Mingwu Zheng, Xin Tao, Wenliang Zhao, Wenzhao Zheng, Jie Zhou, Jiwen Lu, et al. Diffmoe: Dynamic token selection for scalable diffusion transformers. *arXiv preprint arXiv:2503.14487*, 2025.

Oriane Siméoni, Huy V Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, Cijo Jose, Vasil Khalidov, Marc Szafraniec, Seungeun Yi, Michaël Ramamonjisoa, et al. Dinov3. *arXiv preprint arXiv:2508.10104*, 2025.

Sainbayar Sukhbaatar, Arthur Szlam, and Rob Fergus. Learning multiagent communication with backpropagation. In *Advances in Neural Information Processing Systems*, volume 29, pp. 2244–2252, 2016. URL https://papers.nips.cc/paper/6398-learning-multiagent-communication-with-backpropagation.

Sainbayar Sukhbaatar, Olga Golovneva, Vasu Sharma, Hu Xu, Xi Victoria Lin, Baptiste Rozière, Jacob Kahn, Daniel Li, Wen-tau Yih, Jason Weston, and Xian Li. Branch-train-MiX: Mixing expert LLMs into a mixture-of-experts LLM. *arXiv preprint arXiv:2403.07816*, 2024.

Homer Rich Walke, Kevin Black, Tony Z Zhao, Quan Vuong, Chongyi Zheng, Philippe Hansen-Estruch, Andre Wang He, Vivek Myers, Moo Jin Kim, Max Du, et al. Bridgedata v2: A dataset for robot learning at scale. In *Conference on Robot Learning*, pp. 1723–1736. PMLR, 2023.

Binglu Wang, Lei Zhang, Zhaozhong Wang, Yongqiang Zhao, and Tianfei Zhou. Core: Cooperative reconstruction for multi-agent perception. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 8676–8686. IEEE, 2023.

Tonghan Wang, Heng Dong, Victor Lesser, and Chongjie Zhang. ROMA: Multi-agent reinforcement learning with emergent roles. In *International Conference on Machine Learning (ICML)*, 2020.

Yan Wang, Yitao Xu, Nanhan Shen, Jinyan Su, Jimin Huang, and Zining Zhu. The illusion of specialization: Unveiling the domain-invariant " standing committee" in mixture-of-experts models. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 14601–14618, 2026.

Chen Wei, Haoqi Fan, Saining Xie, Chao-Yuan Wu, Alan Yuille, and Christoph Feichtenhofer. Masked feature prediction for self-supervised visual pre-training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

Yujie Wei, Shiwei Zhang, Hangjie Yuan, Yujin Han, Zhekai Chen, Jiayu Wang, Difan Zou, Xihui Liu, Yu Liu, Hongming Shan, et al. Routing matters in moe: Scaling diffusion transformers with explicit routing guidance. In *International Conference on Learning Representations*, volume 2026, pp. 23381–23405, 2026.

Mitchell Wortsman, Gabriel Ilharco, Samir Yitzhak Gadre, Rebecca Roelofs, Raphael Gontijo-Lopes, Ari S. Morcos, Hongseok Namkoong, Ali Farhadi, Yair Carmon, Simon Kornblith, and Ludwig Schmidt. Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time. In *International Conference on Machine Learning (ICML)*, 2022.

Gui-Song Xia, Jingwen Hu, Fan Hu, Baoguang Shi, Xiang Bai, Yanfei Zhong, Liangpei Zhang, and Xiaoqiang Lu. Aid: A benchmark data set for performance evaluation of aerial scene classification. *IEEE Transactions on Geoscience and Remote Sensing*, 55(7):3965–3981, 2017.

Hancheng Ye, Zhengqi Gao, Mingyuan Ma, Qinsi Wang, Yuzhe Fu, Ming-Yu Chung, Yueqian Lin, Zhijian Liu, Jianyi Zhang, Danyang Zhuo, and Yiran Chen. Kvcomm: Online cross-context kv-cache communication for efficient llm-based multi-agent systems. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pp. 17882–17928. Curran Associates, Inc., 2025. doi: 10.52202/085713-0605. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/1a074a28c3a6f2056562d00649ae6416-Paper-Conference.pdf.

Chen Zheng, Yuhang Cai, Deyi Liu, Jin Ma, Yiyuan Ma, Yuan Yang, Jing Liu, Yutao Zeng, Xun Zhou, and Siyuan Qiao. GatePro: Parameter-free expert selection optimization for mixture-of-experts models. *arXiv preprint arXiv:2510.13079*, 2025.

Gaoyue Zhou, Hengkai Pan, Yann LeCun, and Lerrel Pinto. Dino-wm: world models on pre-trained visual features enable zero-shot planning. In *Proceedings of the 42nd International Conference on Machine Learning*, ICML'25. JMLR.org, 2025.

Yanqi Zhou, Tao Lei, Hanxiao Liu, Nan Du, Yanping Huang, Vincent Y. Zhao, Andrew M. Dai, Zhifeng Chen, Quoc V. Le, and James Laudon. Mixture-of-experts with expert choice routing. In *Advances in Neural Information Processing Systems*, volume 35, pp. 7103–7114, 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/2f00ecd787b432c1d36f3de9800728eb-Abstract-Conference.html.

A Implementation details

A.1 Agent architectures

All agents are small ViT networks (Dosovitskiy et al., 2020). Table 4 lists the three sizes. Communication layers are the layers in which a requester's queries may attend to a helper's keys and values. In all other layers every agent attends only to itself.

Table 4: Architecture details of agents of different sizes.

	S	B	L
Model dim	512	768	1024
Number of attention heads	8	12	16
Number of layers	4	6	8
Feed-forward hidden dimension	2048	3072	4096
Dropout	0.05	0.05	0.05
Communication layers	[1, 3]	[1, 2, 4, 5]	[1, 2, 3, 5, 6, 7]
Number of parameters	13.7M	43.1M	103M

A.2 Routers

Table 5 makes explicit what each router receives and how it is trained, since this determines what each regime can and cannot exploit. All learned routers are two-layer MLP heads with their own parameters: LayerNorm, a linear layer from the agent's model dimension d to d, GELU, and a linear layer to K logits, i.e., 0.27M, 0.60M, and 1.06M parameters for S, B, and L agents. The logits are divided by the temperature $\tau = 1.25$ before the softmax. In the central regime the head is applied to the mean of the *visible* DINO tokens of the masked input, so it has exactly the information available to an agent. In the distributed regimes each agent a_q owns a head r_q applied to the mean of that agent's queries in its communication layers. The router therefore reuses the agent's (LoRA-adapted) feature computation, but the MLP parameters are not shared between agents. Distributed delegation and DISCO read the queries of the *same* layers, so that the two regimes differ only in the router's training signal and not in what the router sees. The K-means router clusters the `[CLS]` features of the *uncorrupted* training images offline (computed with a frozen DINOv2 ViT-B/14, i.e., an encoder different from the DINOv3 target space). It is the only router with access to unmasked inputs and is used as a positive control.

Table 5: Information and training signal of each router.

Regime	Router input	Parameters	Training signal
Random	none	none (fixed random split of the training set)	none
K-means	DINOv2 `[CLS]` feature of the *clean* image	none (offline clustering, K clusters)	none
Learned central	mean of visible DINO tokens of the masked input	one shared 2-layer MLP	REINFORCE, reward $= -$ helper loss
Distr. delegation	mean of requester a_q's queries in communication layers	one 2-layer MLP per agent	REINFORCE, reward $= -$ helper loss; requester excluded
DISCO	same as delegation	one 2-layer MLP per agent	REINFORCE, reward $=$ requester's improvement $l_{\text{solo}} - l_{\text{helped}}$; requester excluded

A.3 Training details

All models are pretrained on ImageNet100 for 100 epochs with 64 images per GPU on four GPUs (global batch size 256) using AdamW (Loshchilov & Hutter, 2017) with learning rate 10^{-4} and

weight decay 10^{-4}. The pretrained model is a single agent of the target size trained with the same objective as the populations (a random 30% visible window, cosine loss on the hidden positions) and with all of its weights trainable. AdamW uses $\beta = (0.9, 0.999)$ and $\epsilon = 10^{-8}$; the learning rate is constant, without warm-up or decay, and gradients are clipped to norm 1.0. Images are augmented with a random resized crop (scale 0.6–1.0) to 224×224 and a horizontal flip; DINOv3 ViT-B/16 then yields a 14×14 grid of 196 patch tokens of dimension 768 (the class and register tokens are discarded), which are L2-normalized before an agent sees them. Each agent maps these tokens to its model dimension with a linear layer (the identity for type B, whose model dimension is 768), replaces the hidden positions by a learned mask token, adds a fixed sinusoidal positional encoding over the 196 positions, applies pre-LayerNorm transformer blocks with GELU feed-forward layers and a final LayerNorm, and maps back to 768 dimensions with a linear layer. All populations are trained on 4 NVIDIA GeForce RTX 4090 GPUs. The four data-parallel processes each draw their own requester, so one optimizer step involves four requesters with 64 samples each. Fine-tuning uses the same batch size and optimizer. The populations are fine-tuned on the mixture for 400 epochs using LoRA (Hu et al., 2022a) with rank 64, $\alpha = 8$ and no dropout. LoRA adapters are inserted in the query, key, value, output, and feed-forward projections of the communication layers only (Table 4). All other agent weights, including the non-communication layers, stay frozen, and the routing head is trained in full. The adapters account for 1.18M, 3.54M, and 7.08M parameters per S, B, and L agent and the routing head for 0.27M, 0.60M, and 1.06M, i.e., 1.45M, 4.13M, and 8.13M trainable parameters per agent and 16.5M for a four-agent type-B population. The solo model uses the same adapters (3.54M); its routing head has no decision to make and receives no learning signal. Random and K-means routing train no router, learned central routing trains one head, and the distributed regimes one head per agent. Fine-tuning four type-B agents takes 6.1 hours with DISCO, 5.0 hours with distributed delegation, and 5.3 hours with learned central routing. The solo model takes 4.6 hours. Part of the DISCO overhead is due to additional forward passes used only for logging (e.g., helped reconstructions); Appendix A.5 counts the passes each regime actually needs. All regimes are trained with the same random seed and data order, so that differences between regimes cannot be attributed to initialization or sample order; Section 4.6 varies the seed.

A.4 Data construction

The mixture takes the first 5,999 training and the first 2,190 validation images of every source in the order in which the loader enumerates it (class folders for AID, episode order for the robot datasets, directory order otherwise); 5,999 and 2,190 are the sizes of the smallest source, 102 Flowers. The two robot datasets are stored as episodes. Training and validation use disjoint sets of episodes, an episode is never split across the two, and only every fifth frame of an episode is used, so the 5,999 training frames come from 659 BridgeDataV2 and 559 RT-1 episodes and the 2,190 validation frames from 245 and 203 episodes. KITTI uses the left colour camera of the object benchmark: its training set for training and its test set for validation, which come from different drives. No near-duplicate filtering beyond this frame subsampling was applied, so strided frames of one episode remain similar within a split; this lowers the effective sample size within a split but does not leak across splits. The imbalanced mixture takes the first 4K, 4K, 8K, 9K, 7K, and 4K training images of the same sources. The validation split is the only evaluation split of this study.

A.5 Compute per sample

Table 6 counts the agent passes that each regime needs per training sample and per evaluated sample. Every regime first runs the frozen DINOv3 encoder once per image. Solo, random, and K-means routing train one agent per sample. Learned central routing adds a router. Distributed delegation needs a forward pass of the requester, because its router reads the requester's queries, before the helper is trained. DISCO additionally trains the requester on its helped reconstruction. At evaluation time, delegation runs two agents in the distributed regimes (the requester for routing, the helper for reconstruction) and collaboration three, plus the transfer of the helper's keys and values in four layers. A four-agent population has four times the parameters of the solo model but the same number of active parameters per forward pass. A parameter- or FLOP-matched single model is not part of this study; the random-routing control shows only that four times the parameters do not help by themselves under the fixed sample budget.

Table 6: Agent passes per sample for a four-agent type-B population (fwd/bwd: forward and backward passes). Evaluation counts the forward passes of the delegation and collaboration operating points.

	Training, per sample		Evaluation, agent fwd	
Regime	agent fwd	agent bwd	Deleg.	Collab.
Solo	1	1	1	–
Random, K-means	1	1	1	2
Learned central	1 (+router)	1 (+router)	1	2
Distr. delegation	2	1 (+router)	2	3
DISCO	3	2 (+router)	2	3

A.6 METHOD

Algorithm 1 One DISCO training step for a population $\{a_i, r_i\}_{i=1}^{K}$

Sample a batch $\{x_i = (\hat{z}_i, z_i)\}_{i=1}^{b}$ and a requester index $q \sim \mathrm{Uniform}\{1, \dots, K\}$
For each sample i, draw a helper $j \sim r_q(\hat{z}_i)$ with $r_q^{(q)} \equiv 0$ (no self-selection)
Helpers: compute $l(x_i, a_j)$ and their keys/values K_j^l, V_j^l in communication layers
Requester: compute $l_{\mathrm{solo}}(x_i, a_q)$ and $l_{\mathrm{helped}}(x_i, a_q, a_j)$ using $\mathrm{sg}[K_j^l], \mathrm{sg}[V_j^l]$
Router reward $R_i \leftarrow \mathrm{sg}[\, l_{\mathrm{solo}} - l_{\mathrm{helped}} \,]$; update the requester's baseline $\bar{R}_q$
Update: helpers' LoRA with $\mathcal{L}_{\mathrm{rec}}$; requester's LoRA with $\mathcal{L}_{\mathrm{helped}}$; r_q with $\mathcal{L}_{\mathrm{router}} + \lambda_{\mathrm{KL}}\mathcal{L}_{\mathrm{KL}}$

We use DINOv3 ViT-B/16 (Siméoni et al., 2025) as the frozen encoder, normalize its features before feeding them to the agents, and apply the cosine distance loss only on the masked part of the feature sequence.

During training the helper index is sampled from the distribution predicted by the requester's router, with logits divided by a temperature $\tau = 1.25$. To foster specialization, the requester is excluded from its own routing distribution: its probability is set to 0 and the remaining probabilities are renormalized. Each requester keeps its own exponential-moving-average baseline of improvements with momentum 0.95. The loss weights are $\lambda_{\mathrm{rec}} = 1.25$, $\lambda_{\mathrm{router}} = 1.0$, and $\lambda_{\mathrm{helped}} = 1.0$. All learned routers are regularized with a KL divergence between the marginal routing distribution and the uniform distribution,

$$\mathcal{L}_{\mathrm{KL}} = \sum_{j=1}^{K} \bar{r}^{(j)} \log\big[K \cdot \bar{r}^{(j)}\big], \text{ where } \bar{r}^{(j)} = \frac{1}{b} \sum_{i=1}^{b} r_q^{(j)}(\hat{z}_i) \tag{5}$$

is the marginal, over the current batch, of the routing distribution of the active router r_q (the central router, or the requester's router in the distributed regimes). It is added with weight $\lambda_{\mathrm{KL}} = 0.02$ in all learned-routing regimes. This is the standard load-balancing mechanism of MoE training. As the random-routing control shows, balance alone does not produce specialization; Section 4.6 and Appendix A.8 ablate it.

A.7 SPECIALIZATION DYNAMICS OF THE FIVE ROUTING REGIMES

Figure 6 shows the training-time validation curves of all regimes. Both distributed regimes start from identical agents, so neither router has anything to learn from until the symmetry breaks. What differs is how quickly each reward reflects the break. Delegation is rewarded by the helper's standalone loss, which separates agents as soon as the random batches they are trained on differ, so its routers get a usable signal almost immediately. The reward of DISCO is a difference between two losses of the *same* agent, the requester's reconstruction with and without the message, and that difference only becomes informative once the helpers have begun to differ *and* the requester has learned to read their keys and values. It is therefore smaller and noisier early in training, which is what delays the onset of specialization (72 against 16 epochs to $\mathrm{Sp} = 0.8$). In the last 20 fine-tuning epochs the curves fluctuate with a standard deviation of 0.0005–0.0009 (R-Top5-Best), 0.0007–0.0018 (R-Top5-Worst), 0.001–0.005 (Sp), and below 0.004 (U) for every regime, which we use as the scale of the evaluation

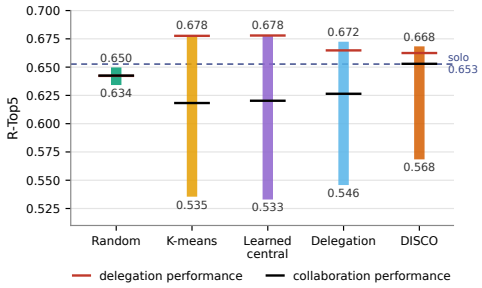


Figure 3: Performance envelopes of the routing regimes (values of Table 2). Bars span R-Top5-Worst to R-Top5-Best; the red and black marks are the delegation and the collaboration performance.

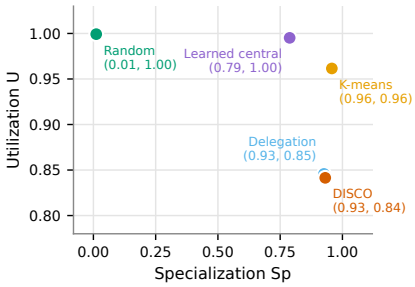


Figure 4: Specialization Sp versus utilization U of the same populations. Random routing is balanced but not specialized; the decentralized regimes are as specialized as the K-means split, at somewhat lower balance.

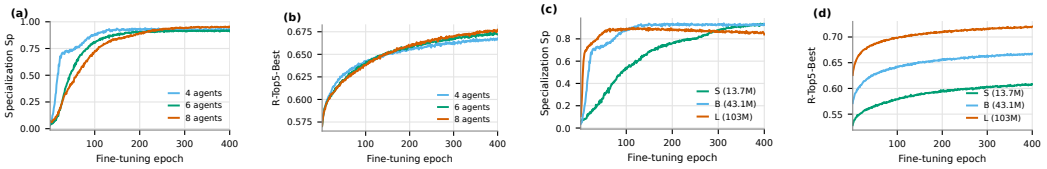


Figure 5: Scaling of DISCO populations during training (training-time validation): Sp and R-Top5-Best. (a,b) Populations of 4, 6, and 8 type-B agents: larger populations specialize later and end higher. (c,d) Populations of four S, B, or L agents: more capable agents specialize earlier and perform better. Discussed in Section 4.4.

noise in Section 4. Table 7 evaluates the regular checkpoints of the DISCO and delegation populations with the protocol of Appendix A.10: the helped score of DISCO rises with the experts' sharpening, while that of delegation, whose agents never learned to read a helper, stays flat. Moreover, the routers agree on the expert for 98% of the inputs from the earliest saved checkpoint on, so agreement is established long before specialization has finished deepening.

Table 7: Routing agreement and specialization along training for the 4-agent DISCO and delegation populations (checkpoints every 50 fine-tuning epochs from epoch 150, one draw of crops and masks, 13,140 validation images). Agree: fraction of inputs on which every non-expert requester's most probable helper is the expert. Agreement is complete by epoch 150 and stays at 98%, while the populations keep specializing (Worst falls) and, in DISCO only, the helped requester keeps improving (Collab. and Worst+h. rise). Sp and U over the same checkpoints are in Figure 6.

Epoch	Regime	Agree	Best	Worst	Collab.	Worst+h.
150	DISCO	**0.985**	0.651	**0.587**	**0.638**	**0.629**
	Delegation	0.981	**0.654**	0.562	0.623	0.604
200	DISCO	**0.982**	0.657	**0.584**	**0.642**	**0.634**
	Delegation	0.971	**0.660**	0.558	0.625	0.603
250	DISCO	**0.982**	0.661	**0.579**	**0.646**	**0.637**
	Delegation	**0.982**	**0.665**	0.554	0.626	0.601
300	DISCO	**0.981**	0.665	**0.576**	**0.649**	**0.641**
	Delegation	0.980	**0.667**	0.552	0.626	0.597
350	DISCO	**0.980**	0.666	**0.572**	**0.651**	**0.642**
	Delegation	0.979	**0.669**	0.548	0.626	0.595
400	DISCO	**0.981**	0.668	**0.568**	**0.653**	**0.644**
	Delegation	0.980	**0.672**	0.546	0.626	0.593

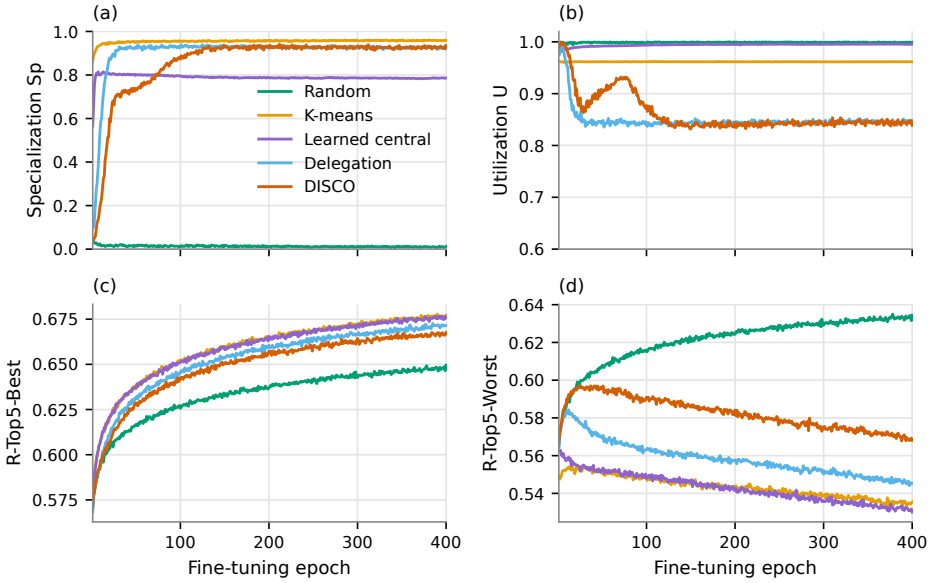


Figure 6: Specialization Sp (a), utilization U (b), R-Top5-Best (c) and R-Top5-Worst (d) during fine-tuning for the five routing regimes of Table 2 (training-time validation, 4 type-B agents). K-means is specialized from the first epoch by construction; the learned central router settles at $\mathrm{Sp} \approx 0.79$ within a few epochs; delegation passes $\mathrm{Sp} = 0.8$ after 16 epochs and DISCO after 72. Utilization starts near 1 everywhere and stays there under random and learned central routing. It falls only in the two distributed regimes, where responsibility concentrates on three of the four agents, and in DISCO it rebounds for a few tens of epochs before settling at the level of delegation ($\mathrm{U} \approx 0.84$). Panels (c) and (d) show the price of specialization: Best rises together with Sp, while Worst falls for every specialized population and falls least for DISCO. Random routing stays at $\mathrm{Sp} \approx 0.01$ and keeps the highest Worst.

Table 8: Final checkpoints of the scaling study evaluated with the protocol of Appendix A.10 (mixture validation set, three draws). With more agents the oracle Best of DISCO rises and stays 0.007–0.010 below a K-means split of the same size, whose collaboration score is 0.03–0.05 lower. The helped requester of the 8-agent population exceeds the solo model (0.655 vs. 0.653). Agreement decreases with population size (98%, 98%, 88%) because each router is trained on fewer batches. Capacity moves every score in the same direction: from four S to four B to four L agents, Best rises 0.607, 0.668, 0.721, Collab. 0.591, 0.653, 0.709 and Worst+h. 0.580, 0.644, 0.704, and the help closes more of the gap to the expert as the agents grow (Worst+h. is 0.027 below Best for S, 0.024 for B and 0.017 for L). Specialization stays high for S and B ($\mathrm{Sp} \approx 0.93$) and is lower for L (0.84) at a higher utilization (0.91). Each population is initialized from a pretrained model of its own size.

Population	Sp	U	Best	Worst	Deleg.	Collab.	Worst+h.	Agree
DISCO, 4 B agents	**0.931**	0.841	0.668	**0.568**	0.662	**0.653**	0.644	**0.983**
DISCO, 6 B agents	0.914	**0.998**	**0.674**	**0.579**	**0.666**	0.652	**0.645**	**0.982**
DISCO, 8 B agents	**0.953**	**0.936**	**0.678**	0.543	**0.673**	**0.655**	**0.647**	0.881
K-means, 4 B agents	**0.957**	**0.962**	0.678	**0.535**	0.678	**0.618**	**0.582**	–
K-means, 6 B agents	0.887	**0.935**	**0.681**	**0.532**	**0.679**	**0.607**	**0.573**	–
K-means, 8 B agents	**0.928**	0.912	**0.685**	0.530	**0.683**	0.606	0.572	–
DISCO, 4 S agents	**0.926**	0.843	0.607	0.528	0.602	0.591	0.580	**0.979**
DISCO, 4 L agents	0.841	**0.909**	**0.721**	**0.612**	**0.715**	**0.709**	**0.704**	0.939

A.8 INDEPENDENT FINE-TUNING SEEDS AND PROTOCOL ABLATIONS

Table 9 lists all replication and ablation runs. Every run uses the configuration of its regime in Section 3.3 with a new seed, which changes the LoRA and router initialization, the data order, the requester and helper draws, and the masks; the same seed gives the same data order in every regime, so rows with equal seeds are paired. Runs last 100 fine-tuning epochs, a quarter of the budget of Table 2, and the final checkpoint is evaluated with the protocol of Appendix A.10 on one draw of crops and masks.

Seeds. Training-time validation places the onset of specialization (first validation with $\mathrm{Sp} \geq 0.8$) at fine-tuning epoch 28, 28, and 32 for DISCO and 16, 28, and 16 for delegation. Two DISCO populations and one delegation population use all four agents (expert shares 17–33%); the others leave one agent with 2% (DISCO, seed 2) or 9–10% (delegation, seeds 1–2) of the inputs. Whether an agent is left idle is thus a property of the run, not of the regime. The same-requester interventions of Table 3 hold for every seed: a DISCO requester scores 0.614–0.620 alone, 0.631–0.632 with its selected helper, 0.622–0.624 with a random other agent, 0.600–0.606 with zero keys and values, and 0.395–0.398 with a shuffled message (delegation: 0.602–0.608, 0.621–0.623, 0.607–0.611, 0.592–0.598, 0.404–0.405).

Ladder. The ordering of Section 4.1 replicates at this budget: three solo seeds score 0.633–0.634, two random-routing populations 0.628 ($\mathrm{Sp} = 0.02$, $\mathrm{U} = 1.00$), the distributed regimes 0.645–0.646, and two learned central routers 0.651–0.652 ($\mathrm{Sp} = 0.70$–0.81, $\mathrm{U} = 1.00$). The helped DISCO requester (0.631–0.632) is 0.002–0.003 below the solo model at this budget (on par with it at the full budget, Table 2). On ImageNet100 the solo seeds score 0.654–0.655, DISCO's collaboration 0.657–0.658 and its delegation 0.656, delegation's collaboration 0.655–0.656, and learned central routing 0.651–0.654 with communication against 0.654 without, so DISCO remains the only regime whose collaboration exceeds both the solo model and its own delegation.

Reward versus channel. The cell "delegation reward + communication training" trains the requester on its helped loss but rewards its router with the helper's loss, as in delegation; "random routing + communication training" draws the helper uniformly among the other agents and trains no router (its Deleg. column is the score of an average agent and its Collab. column the random-helper collaboration). The first reproduces the helped scores of DISCO (collaboration 0.631, Worst+help 0.624) together with delegation's specialization and routing ($\mathrm{Sp} = 0.86$, $\mathrm{U} = 0.93$, agreement 82%, onset after 20 epochs, least-used agent 8%). The second never breaks the symmetry ($\mathrm{Sp} = 0.00$, $\mathrm{U} = 1.00$, expert shares 24–26%); its agents reach 0.630 as experts and 0.622 as worst agents, and a random helper's message is worth 0.001 to a requester (0.627 vs. 0.626).

Protocol choices. With $\lambda_{\mathrm{KL}} = 0$ the DISCO population collapses onto one agent within the first four fine-tuning epochs and stays there: the routers of three agents send every input to agent 2 and agent 2's own router sends everything to agent 1, so agent 2 is trained on three quarters of the traffic and is the expert on 90% of the inputs (expert shares 9%, 90%, 1%, 0%; $\mathrm{Sp} = 0.08$, $\mathrm{U} = 0.25$). Its Best (0.633) equals the solo model of the same budget and its Worst (0.566) that of a specialized population; a helped requester still profits from the generalist's message (0.628 vs. 0.599 alone, 0.588 with zero and 0.405 with shuffled keys and values), but nothing is divided. The KL term is thus a precondition for the division of labor under local routing, as load balancing is for MoEs; the random-routing control shows that it is not sufficient. With self-selection allowed (the requester's own index is not removed from its routing distribution), specialization still emerges, later and weaker ($\mathrm{Sp} = 0.81$, $\mathrm{U} = 0.99$; Sp passes 0.8 after 72 epochs against 28–32 with exclusion): agents keep the aerial images and faces to single experts and split the two robotics domains and the driving images between two agents. The routers select their own agent on 24% of the inputs, on 91% of those on which their agent is the expert and on 2% of the others, and non-expert requesters select the expert on 93% of the inputs, so the exclusion sharpens and accelerates the division of labor but is not what creates it. The helped scores equal those of DISCO (collaboration 0.633, Worst+help 0.625; a requester alone 0.604, with zero keys and values 0.591, shuffled 0.391).

Table 9: Replication and ablations at a quarter of the training budget (100 fine-tuning epochs, seeded protocol of Appendix A.10, one draw of crops and masks). Seeds 1–3 change the LoRA and router initialization, the data order, the requester and helper draws, and the masks; the same seed gives the same data order in every regime. Mean ± standard deviation over seeds where several seeds exist.

Regime	Seed	Sp	U	Best	Worst	Deleg.	Collab.	Worst+help	Agree
Solo	1	–	–	0.633	0.633	–	–	–	–
Solo	2	–	–	0.634	0.634	–	–	–	–
Solo	3	–	–	0.634	0.634	–	–	–	–
mean ± sd		–	–	0.634 ±.001	0.634 ±.001	–	–	–	–
Random	1	0.018	0.997	0.628	0.617	0.622	0.623	0.619	–
Random	2	0.018	0.999	0.628	0.617	0.622	0.623	0.619	–
mean ± sd		0.018 ±.000	0.998 ±.002	0.628 ±.000	0.617 ±.000	0.622 ±.000	0.623 ±.000	0.619 ±.000	–
Learned central	1	0.808	0.999	0.652	0.557	0.652	0.615	0.600	–
Learned central	2	0.700	0.998	0.651	0.550	0.651	0.615	0.598	–
mean ± sd		0.754 ±.076	0.999 ±.001	0.652 ±.001	0.553 ±.005	0.651 ±.001	0.615 ±.000	0.599 ±.002	–
Distr. delegation	1	0.863	0.943	0.646	0.565	0.640	0.622	0.607	0.820
Distr. delegation	2	0.855	0.932	0.646	0.571	0.640	0.623	0.608	0.818
Distr. delegation	3	0.950	0.964	0.646	0.566	0.640	0.621	0.607	0.985
mean ± sd		0.889 ±.053	0.946 ±.016	0.646 ±.000	0.568 ±.003	0.640 ±.000	0.622 ±.001	0.607 ±.001	0.874 ±.096
DISCO	1	0.948	0.963	0.644	0.601	0.639	0.631	0.624	0.986
DISCO	2	0.919	0.846	0.645	0.585	0.640	0.632	0.625	0.980
DISCO	3	0.942	0.963	0.645	0.597	0.639	0.631	0.625	0.986
mean ± sd		0.936 ±.015	0.924 ±.068	0.645 ±.000	0.594 ±.009	0.639 ±.001	0.631 ±.001	0.625 ±.001	0.984 ±.004
Delegation reward + comm. training	1	0.855	0.926	0.645	0.586	0.640	0.631	0.624	0.818
Random routing + comm. training	1	0.002	1.000	0.630	0.622	0.626	0.627	0.625	0.004
DISCO, $\lambda_{\mathrm{KL}} = 0$	1	0.084	0.255	0.633	0.566	0.631	0.628	0.625	0.905
DISCO, self-selection allowed	1	0.812	0.995	0.647	0.572	0.646	0.633	0.625	0.915

A.9 Which domains each agent takes, and where help pays off

In this section, we look at what the agents divide and at what the help actually restores. Figure 7 makes the division concrete: on one validation image per domain, the expert reconstructs structures that the highest-loss agent misses entirely, and the same non-expert recovers most of them once it reads the expert's keys and values, which is the qualitative counterpart of the Worst+help column of Table 2. Figure 8 then shows which domains each agent takes in every specialized population, Table 10 breaks the scores down by domain, and Figure 9 shows how much of their attention requesters spend on the helper's tokens, over the validation set (left) and for a single input (right).

Table 10: Per-domain R-Top5 on the mixture validation set (2,190 images per domain). Best is the expert's score, Worst the highest-loss agent's, Collab. the score of a random requester helped by its selected helper, Worst+help that of the worst agent helped. Solo scores show that the domains differ widely in difficulty. The collaboration gain of DISCO over delegation is largest on faces, followed by the two robotics domains, and is absent on aerial images, the domain on which delegation leaves its idle agent.

Regime / measure	AID	KITTI	Bridge	RT-1	CelebA	Flowers
Solo	0.429	0.741	0.694	0.724	0.788	0.540
K-means, Best (expert)	**0.462**	0.757	**0.733**	**0.750**	**0.805**	**0.563**
Learned central, Best (expert)	0.444	**0.762**	**0.749**	**0.754**	**0.802**	**0.558**
Delegation, Best (expert)	**0.448**	**0.759**	0.726	0.747	**0.802**	0.553
DISCO, Best (expert)	0.442	0.754	0.719	0.742	0.800	0.554
Delegation, Worst	**0.407**	0.664	0.515	0.541	0.666	0.481
DISCO, Worst	0.390	**0.687**	**0.541**	**0.590**	**0.697**	**0.506**
Delegation, Collab.	**0.433**	0.729	0.653	0.683	0.735	0.524
DISCO, Collab.	0.430	**0.743**	**0.692**	**0.722**	**0.787**	**0.543**
Delegation, Worst+help	**0.437**	0.707	0.607	0.639	0.663	0.502
DISCO, Worst+help	0.424	**0.738**	**0.678**	**0.711**	**0.777**	**0.536**

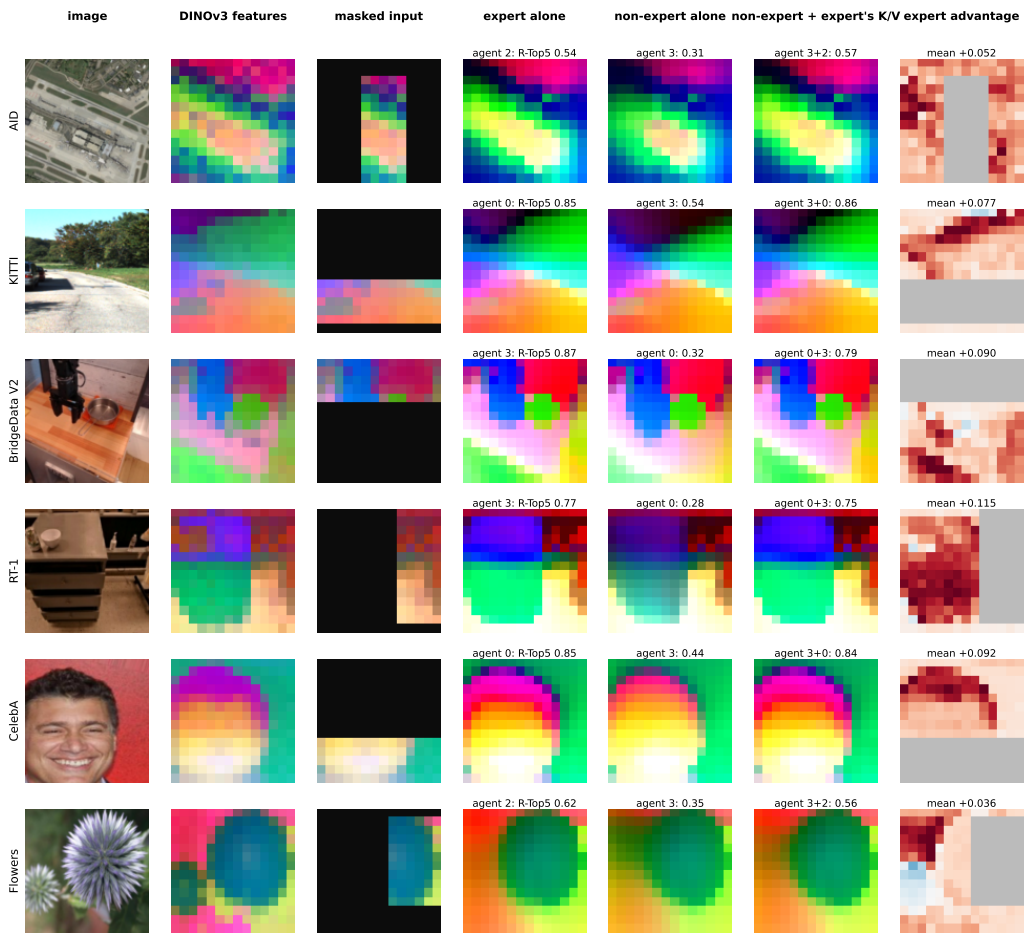


Figure 7: What the expert knows and the non-expert does not. One validation image per domain, 4-agent DISCO population, same crops and masks as in Table 2. DINOv3 patch features are projected to RGB by a PCA fitted on the target features of each image. The masked input shows the 30% visible window. Columns 4–6: reconstructions of the expert (lowest-loss agent) alone, of the highest-loss agent alone, and of the same non-expert reading the expert's keys and values, with their R-Top5. Last column: per-patch reconstruction error (cosine distance to the DINOv3 target) of the non-expert minus that of the expert, on the hidden patches (red: the expert is better; visible window grey; mean in the title). The non-expert misses whole structures, e.g., the pan and gripper in BridgeData V2, the road–vegetation boundary in KITTI, the hairline in CelebA, which the expert reconstructs and the help restores. Examples are those with the largest help gain per domain.

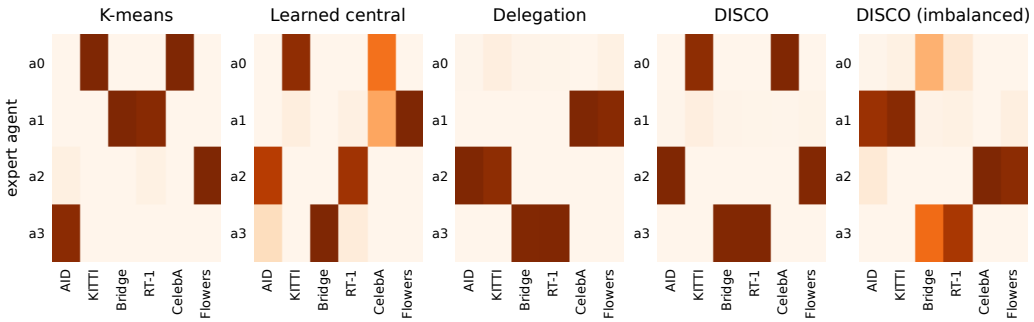


Figure 8: Fraction of each domain's validation images whose expert (lowest standalone loss) is agent a_i, for the specialized regimes of Table 2 and the imbalanced mixture. Delegation and DISCO both split the six domains into three pairs and leave one agent almost idle, which is what lowers their utilization to $\mathrm{U} \approx 0.84$, but the pairs differ: delegation groups aerial with driving and faces with flowers, DISCO groups aerial with flowers and driving with faces, and only the two robotics datasets are kept together by both; the learned central router splits the face images across two agents; K-means separates aerial images from flowers. On the imbalanced mixture the two robotics datasets (8K and 9K training images) are split between two agents instead.

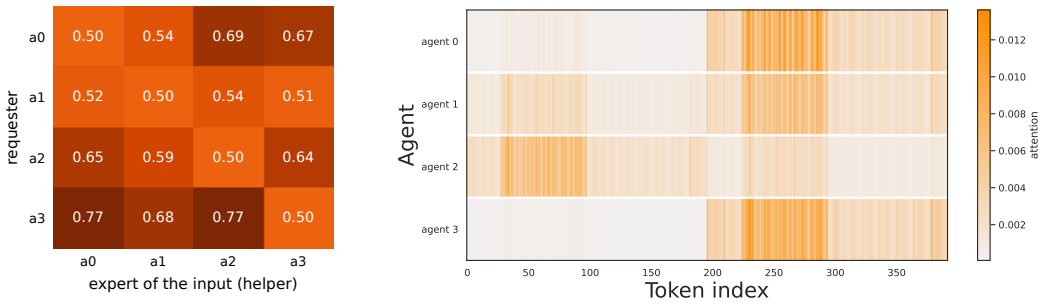


Figure 9: Where the requesters look. **Left:** attention mass that requester a_i (rows) places on the helper's keys and values in the communication layers of the 4-agent DISCO population, on inputs whose expert is agent a_j (columns), averaged over heads, queries, layers, and the validation set. When requester and helper coincide (diagonal) the two copies of the same keys receive half of the mass each, so 0.5 is the reference. The three specialists place 0.64–0.77 of their attention on the expert's tokens, while the almost idle agent a_1 (Figure 8) uses helpers little (0.51–0.54). **Right:** per-token attention of every agent in the last communication layer, for one input whose expert is agent 2 (mean over heads and queries; token indices 0–195 are the requester's own keys, 196–391 the helper's). The expert attends to its own keys; the other agents attend mostly to the expert's.

A.10 EVALUATION PROTOCOL

Sp and U are computed on the validation sets of the six domains (2,190 images each, 13,140 in total) from the standalone losses of the agents on the same masked inputs. The routers play no role in these two metrics. R-Top5-Best and R-Top5-Worst are the expert envelope over agents (Section 3.2). The *delegation* performance evaluates the population with its own router: each validation sample is reconstructed by the router's most probable agent (for a random requester in the distributed regimes, whose own index is excluded). For K-means and random routing the fixed assignment of the sample is used. The *collaboration* performance draws a random requester for each sample and lets it reconstruct with the keys and values of the selected helper. For the held-out evaluation (Section 4.3), the same protocol is applied to the ImageNet100 validation set, which none of the populations saw during specialization. The routing-agreement statistic (Section 4.5) counts validation inputs for which every non-expert requester's most likely helper is the expert $e(x)$.

Metric. Two model-free predictors calibrate R-Top5 on the same inputs: copying the feature of the spatially nearest visible patch into every hidden position scores 0.120, predicting the mean of the visible features scores 0.001, and chance is $5/196 = 0.026$; the pretrained agent before specialization scores 0.563 and the solo model 0.653 (Table 2). The expert is selected by loss and scored by R-Top5. The lowest-loss agent is also the agent with the highest R-Top5 on 71% (DISCO), 73% (delegation), 84% (learned central), and 87% (K-means) of the inputs, but only on 41% under random routing, whose agents are nearly interchangeable. Averaging instead the per-image maximum of R-Top5 over agents gives 0.673, 0.678, 0.683, 0.682, and 0.660 for these five regimes, 0.004–0.010 above the loss-selected Best of Table 2, and the per-image minimum gives 0.560, 0.524, 0.511, 0.511, and 0.624, 0.008–0.024 below the loss-selected Worst. The loss-selected envelope is thus slightly narrower, and the ordering of the specialized regimes is the same under both. The random population is the exception: its R-Top5 maximum (0.660) would exceed the solo model (0.653), because maximizing the reported metric over four nearly interchangeable agents selects evaluation noise. This is why the expert is defined by an independent criterion, the loss, and why we do not report the metric's own extrema as the envelope.

Message interventions. Table 3 uses the same cached inputs and the same requester–helper pairs as Table 2; the shuffled message is the selected helper's keys and values for the next image of the evaluation batch, and the zero message has the same token count.

Seeded protocol and intervals. Validation uses the training-time augmentation (random resized crop and flip) and a random 70% mask per image, so an evaluation depends on the draw. All numbers in Section 4 come from one protocol: three independent draws of crops and masks are fixed per image and shared by all regimes, every agent is run on every image, and every (requester, helper) pair is evaluated, so all comparisons are paired. Point estimates are means over the three draws. Three quantities describe how much a number can move, and they are not interchangeable.

(i) Draw-to-draw spread. Re-drawing the crops and masks and repeating the whole evaluation changes a point estimate on the mixture by at most 0.0005 for Best, Deleg., Collab. and Worst+help, by 0.0012 for Worst, and by 0.005 for Sp. On ImageNet100 the spread of the R-Top5 quantities is at most 0.002.

(ii) Interval on a single entry. Uncertainty from the finite validation set is a paired bootstrap over images (10,000 resamples, an image carrying its three draws). It asks how much one cell of Table 2 would move if the 13,140 images were replaced by another sample of the same size. Because the domains differ widely in difficulty (Table 10), that is the largest of the three: across all regimes and all R-Top5 columns the 95% half-width is 0.002–0.003, and for Sp it reaches 0.007.

(iii) Interval on a difference between two regimes. The same images, crops and masks are used for every regime, so a hard image lowers both scores at once and cancels out of the paired difference. Over the comparisons listed below, the 95% half-width is 0.0002–0.0004 for Best and Deleg., up to 0.0007 for Collab., and up to 0.0015 for Worst and Worst+help, whose per-image values are more dispersed; on ImageNet100 the corresponding half-widths are 0.0005–0.0012. For Sp, which is a single population-level statistic rather than a per-image mean, the half-width on a difference is at most 0.005.

This is why the comparisons in Section 4 are stated as paired differences: a gap of 0.02–0.03 between two regimes is resolved by an interval an order of magnitude narrower, even though the individual entries it is computed from carry a spread of the same order as the gap. The intervals below are exactly these paired differences, which is why they are much tighter than the entries in Table 2. Key differences: random − solo Best -0.003 $[-0.003, -0.003]$; learned central − solo Best $+0.026$ $[+0.025, +0.026]$; DISCO − delegation: Best -0.004 $[-0.004, -0.004]$, Worst $+0.023$ $[+0.022, +0.023]$, Collab. $+0.026$ $[+0.026, +0.027]$, Worst+help $+0.051$ $[+0.050, +0.052]$, Sp $+0.005$ $[+0.000, +0.010]$; DISCO Collab. − solo $+0.0002$ $[0.0000, +0.0004]$.

B LIMITATIONS

This study is a controlled investigation of emergent specialization, not a training recipe, and its evidence has clear boundaries.

Task and modality. All experiments use masked reconstruction of frozen DINOv3 features. We argued in Section 3.2 why this is a reasonable proxy for predictive pretraining, but transfer of the findings to raw-pixel generation, video prediction, language modeling, multimodal learning, or downstream task optimization has not been tested. The key–value communication protocol itself is task-agnostic and applies to any transformer, so the protocol is not the obstacle; the cost of running population-level sweeps in those settings is.

Scale. Populations have at most eight agents of at most 103M parameters, trained on 36K images. The uniform choice of the requester slows specialization as the population grows (Section 4.4). Much larger populations would need a different scheduling of who routes, and the interaction of specialization with web-scale data remains open.

Protocol choices. Communication carries the keys and values of a single helper per sample and is deliberately gradient-free. We did not compare against a differentiable channel, nor against multi-helper or hierarchical communication. A differentiable channel would answer a different question: this study asks whether specialization and usable expertise can emerge when no gradient crosses between agents, and our results show that they can. Domain identity is used as the latent factor for evaluation. Specialization along factors that do not align with dataset boundaries would require other probes.

C IMPACT STATEMENT

This work is foundational research on modular and collaborative neural systems. Its main potential benefit is to improve the efficiency, adaptability, and interpretability of machine learning models by encouraging populations of agents to divide labor across heterogeneous data. Such systems could eventually reduce the need for monolithic models by enabling more targeted use of capacity and by making it easier to analyze which components are responsible for different parts of a distribution.

At the same time, more adaptive multi-agent systems may inherit or amplify risks associated with the domains in which they are deployed. If used in high-stakes applications, emergent specialization could produce uneven performance across domains or user groups, especially when the latent structure of the data reflects sensitive or imbalanced attributes. We therefore view careful measurement of specialization, utilization, robustness, and fairness as important prerequisites before applying such methods outside controlled research settings. The present work does not introduce new datasets or deployment-ready models, and we do not anticipate direct societal harms from the experiments themselves.